

Application and validation of capacitive proximity sensing systems in smart environments



Vom Fachbereich Informatik
der Technischen Universität Darmstadt
genehmigte

DISSERTATION

zur Erlangung des akademischen Grades eines
Doktor-Ingenieurs (Dr.-Ing.)
von

M.Sc. Andreas Braun

geboren in Aschaffenburg, Deutschland

Referenten der Arbeit: Prof. Dr. techn. Dieter W. Fellner
Technische Universität Darmstadt
Prof. Dr. Max Mühlhäuser
Technische Universität Darmstadt

Tag der Einreichung: 24.07.2014
Tag der mündlichen Prüfung: 18.09.2014

Darmstädter Dissertation 2014
D 17

Erklärung zur Dissertation

Hiermit versichere ich die vorliegende Dissertation selbständig nur mit den angegebenen Quellen und Hilfsmitteln angefertigt zu haben. Alle Stellen, die aus Quellen entnommen wurden, sind als solche kenntlich gemacht. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Darmstadt, den 24.07.2014

Andreas Braun

Abstract

Smart environments feature a number of computing and sensing devices that support occupants in performing their tasks. In the last decades there has been a multitude of advances in miniaturizing sensors and computers, while greatly increasing their performance. As a result new devices are introduced into our daily lives that have a plethora of functions. Gathering information about the occupants is fundamental in adapting the smart environment according to preference and situation. There is a large number of different sensing devices available that can provide information about the user. They include cameras, accelerometers, GPS, acoustic systems, or capacitive sensors. The latter use the properties of an electric field to sense presence and properties of conductive objects within range. They are commonly employed in finger-controlled touch screens that are present in billions of devices. A less common variety is the capacitive proximity sensor. It can detect the presence of the human body over a distance, providing interesting applications in smart environments. Choosing the right sensor technology is an important decision in designing a smart environment application. Apart from looking at previous use cases, this process can be supported by providing more formal methods.

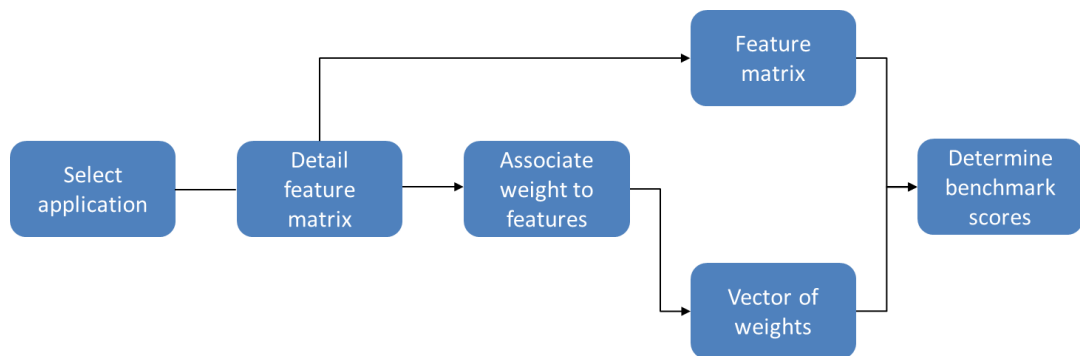
In this work I present a benchmarking model that is designed to support this decision process for applications in smart environments. Previous benchmarks for pervasive systems have been adapted towards sensors systems and include metrics that are specific for smart environments. Based on distinct sensor characteristics, different ratings are used as weighting factors in calculating a benchmarking score. The method is verified using popularity matching in two scientific databases. Additionally, there are extensions to cope with central tendency bias and normalization with regards to average feature rating. Four relevant application areas are identified by applying this benchmark to applications in smart environments and capacitive proximity sensors. They are indoor localization, smart appliances, physiological sensing and gesture interaction. Any application area has a set of challenges regarding the required sensor technology, layout of the systems, and processing that can be tackled using various new or improved methods. I will present a collection of existing and novel methods that support processing data generated by capacitive proximity sensors. These are in the areas of sparsely distributed sensors, model-driven fitting methods, heterogeneous sensor systems, image-based processing and physiological signal processing. To evaluate the feasibility of these methods, several prototypes have been created and tested for performance and usability. Six of them are presented in detail. Based on these evaluations and the knowledge generated in the design process, I am able to classify capacitive proximity sensing in smart environments. This classification consists of a comparison to other popular sensing technologies in smart environments, the major benefits of capacitive proximity sensors, and their limitations. In order to support parties interested in developing smart environment applications using capacitive proximity sensors, I present a set of guidelines that support the decision process from technology selection to choice of processing methods.

Zusammenfassung

Das Forschungsgebiet Smart Environments beschreibt Umgebungen die über eine Vielzahl von Sensoren und Computern erweitert werden. Diese unterstützen Personen, welche in dieser Umgebung agieren, in der Erfüllung verschiedener Aufgaben. Technische Entwicklungen der vergangenen Jahrzehnte führten zu einer zunehmenden Miniaturisierung von Sensoren und Computern, während die Rechenleistung beziehungsweise zw. erreichbare Auflösung stark anstieg. In Folge nutzen wir zunehmend neue technische Geräte, welche über eine hohe Zahl von Funktionen verfügen. Eine der wichtigsten Aufgaben von Smart Environments ist die Sammlung von Informationen über Situation und Präferenz der Nutzer, welche dazu genutzt werden können die Umgebung anzupassen. Es existieren diverse Sensortechnologien, die es ermöglichen derartige Information zu gewinnen. Beispiele hierfür sind Kameras, Beschleunigungssensoren, GPS, akustische Systeme oder auch kapazitive Sensoren. Die letztgenannten messen die Eigenschaften eines elektrischen Feldes um Präsenz und Eigenschaften von leitfähigen Objekten zu bestimmen, welche in dieses Feld eintreten. Diese Technologie ist die Basis für fingerkontrollierte Touchscreens, die bereits in mehreren Milliarden Geräten verbaut wurden. Eine weniger bekannte Variante sind kapazitive Abstandssensoren. Diese sind in der Lage die Präsenz von menschlichen Körperteilen über eine gewisse Distanz zu erkennen. Dies ermöglicht interessante Anwendungen in Smart Environments. Bei der Erstellung von Anwendungen in dieser Domäne ist die Wahl passender Sensoren eine der wichtigsten Entscheidungen. Dies wird bislang primär durch eine Analyse bestehender Lösungen realisiert, welche eine gewisse Ähnlichkeit zum gewählten Ansatz haben. Ziel dieser Arbeit ist die Einordnung von kapazitiven Abstandssensoren im Bereich Smart Environments. Hierzu wird ein Benchmarking-Modell erstellt, das es ermöglicht über eine Auswahl von Sensoreigenschaften eine Eignung für bestimmte Anwendungsgebiete zu errechnen. Dies ermöglicht es verschiedene Nutzungsszenarien für kapazitive Abstandssensoren in Smart Environments zu finden, bzw. zu verifizieren. Indem eine Zahl von Prototypen für diese Szenarien entwickelt wird, ist es mir zum einen möglich neue Datenverarbeitungsmethoden zu realisieren, und zum anderen die erwähnte Klassifikation vorzunehmen. Diese besteht aus einem Vergleich von kapazitiven Abstandssensoren mit anderen populären Sensortechnologien, einer Diskussion der spezifischen Vor- und Nachteile, sowie einer Sammlung von Richtlinien, welche Entwickler dabei unterstützen, Anwendungen für Smart Environments zu realisieren.

Die Diskussion der verwandten Arbeiten ist in vier verschiedene Bereiche gegliedert und in Kapitel 2 zu finden. Zunächst gebe ich eine Einführung in die generelle Methodik kapazitiver Feldmessungen. Dies beinhaltet die physikalischen Grundlagen, eine Diskussion verschiedener Messmethoden und Anwendungsbereiche, den Einfluss von Material und Geometrie auf die Messwerte, sowie eine Einführung in relevante Methoden zur Datenverarbeitung. Letzteres, betrachtet sowohl Varianten zur Verarbeitung der Rohdaten, als auch höherwertige Verarbeitungsmethoden, wie z.B. Objektverfolgung und Datenfusion. Der zweite Abschnitt führt verschiedene Anwendungen ein, die mit kapazitiven Abstandssensoren realisiert wurden. Diese umfassen klassische Arbeiten, wie den Theremin, ein elektronisches Musikinstrument des frühen zwanzigsten Jahrhunderts [Gli00], oder die Arbeiten am MIT Media Lab in den 1990ern [Smi99]. Des Weiteren, werden neuere Arbeiten der Forschungsgruppen an der LMU München [WKBS07], VTT Finnland [VMV09], Disney Research [SPH12] eingeführt, sowie relevante Einzelprojekte wie SmartSkin von Jun Rekimoto [Rek02], oder ein 3D Interaktionssystem von Le Goc et al., basierend auf transparenten Elektroden [LTI*14]. Im dritten Abschnitt gebe ich einen Überblick populärer Sensortechnologien, die in Smart Environments genutzt werden, sowie beispielhafte Anwendungen. Eulerian Video Magnification ist eine Methode für RGB Kameras, die zeitliche Farbvariationen verstärkt. Dies

kann genutzt werden, um die aktuelle Herzfrequenz zu erkennen [WRS*12]. Infrarotkameras ermöglichen die berührungslose Interaktion, z.B. zur Kontrolle von Anwendungen in einer Küche [Pan12]. Ultraschallsensoren, emittieren hochfrequente mechanische Wellen, die z.B. zur Lokalisierung genutzt werden können [PCB00]. Mikrofone nehmen akustische Umgebungsgeräusche auf, um zum Beispiel die aktuell sprechende Person zu erkennen [ZFBZ08]. Radiowellen können vielfältig genutzt werden, z.B. auch zur Erkennung von Ganzkörpergestik [PGGP13]. Im letzten Abschnitt stelle ich relevante Anwendungen in Smart Environments vor, die potentiell mit kapazitiver Abstandssensorik realisiert werden könnten. Diese werden gruppiert in Innenlokalisierung, Gesteninteraktion, physiologische Sensorik, Aktivitätserkennung, Smart Appliances, mobile Geräte und autonome Systeme. In jedem dieser Bereiche werden relevante Anwendungen der letzten Jahre vorgestellt. Diese verwandten Arbeiten dienen als Basis für das Benchmarking-Modell, für welches ein umfassender Überblick von Anwendungsgebieten und Sensortechnologien notwendig ist.



Benchmarking Prozess

In Kapitel 3 wird das Benchmarking-Modell eingeführt. Dieses ermöglicht die Errechnung eines Wertes, der einen Hinweis gibt, inwiefern eine spezifische Technologie für ein Anwendungsgebiet geeignet ist. Zum einen ist es so möglich, geeignete Anwendungen für eine Technologie zu finden, zum anderen können auch umgekehrt Technologien für eine spezifische Anwendung evaluiert werden. Zunächst wird Benchmarking als Methode eingeführt, insbesondere mit Verweis auf bestehende Arbeiten im Bereich Pervasive Computing [RAMB*05]. Die relevanten Sensoreigenschaften können in drei Gruppen sortiert werden. Diese sind Sensorleistung, Umgebungscharakteristiken und „pervasive metrics“, welche Eigenschaften beschreiben, die in Kooperation mit Smart Environments relevant sind. Für jede Eigenschaft existieren fünf mögliche Bewertungen, welche als Gewichtung in die spätere Berechnung einfließen. Als nächstes wird das eigentliche Benchmarking-Modell eingeführt, sowie dessen mathematische Herleitung. Als Erweiterung werden verschiedene Methoden zur Normalisierung vorgeschlagen, sowie eine Kompensation der Tendenz von Nutzern, mittlere Bewertungen zu bevorzugen. Über eine Analyse der Trefferhäufigkeit verschiedener Stichworte bei Suche in der ACM Digital Library, sowie Google Scholar, wird das Benchmarking-Modell verifiziert. Anschließend wurde die Methodik auf kapazitive Abstandssensorik angewandt, um diejenigen Anwendungsdomänen in Smart Environments zu bestimmen, die am relevantesten für diese Technologie sind. Diese Domänen sind Innenlokalisierung, Smart Appliances, physiologische Sensorik, sowie Gesteninteraktion.

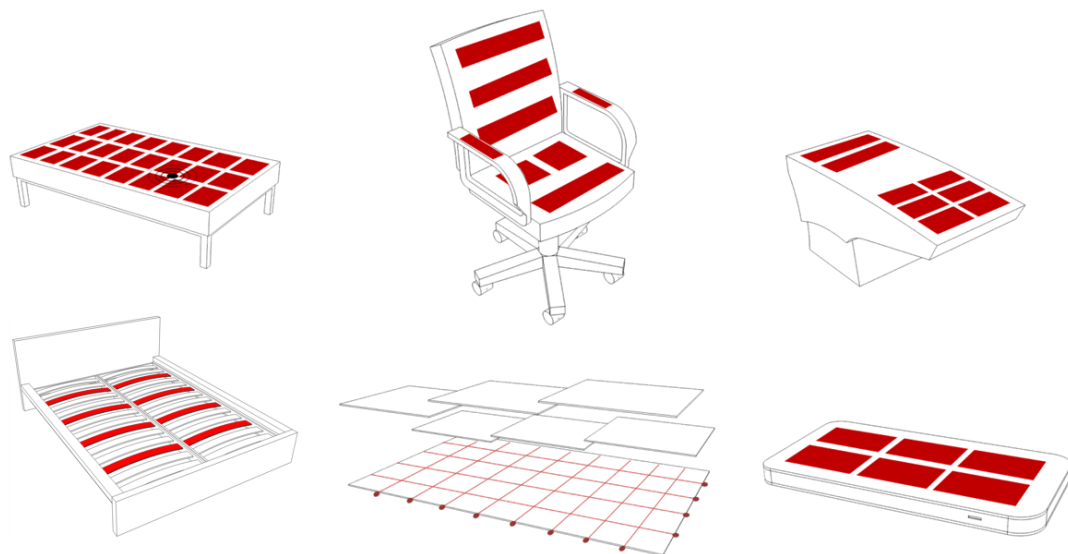
Kapitel 4 diskutiert spezifische Anwendungen in den gefundenen Domänen. Im nächsten Schritt werden charakteristische Probleme der Anwendungen identifiziert um verschiedene Bereiche zu identifizieren, in welchen neue, bzw. erweiterte Datenverarbeitungsmethoden relevant sein könnten. Anwendungen, sowie charakteristische Herausforderungen sind in der folgenden Tabelle dargestellt. Um diese Herausforderungen zu meistern, werden Datenverarbeitungsmethoden in fünf verschiedenen Bereichen dargestellt. Für verteilte Sensorsysteme

Herausforderungen der verschiedenen Anwendungsszenarien für kapazitive Abstandssensoren

Anwendungsszenario	Herausforderungen
Kapazitive Bodensensoren	Weit verteilte Sensoren, geometrisches Elektrodenlayout
Posenerkennender Stuhl	Mehrkörpermodelle, Elektrodenmaterial
Belegungserkennendes Bett	Einkörpermodell, Bewegungserkennung
Gestenerkennende Armlehne	Heterogene kapazitive Systeme
Atemfrequenzmessung	Frequenzraumanalyse
Schlafphasenerkennung	Langzeitbewegungsanalyse
Mikrogesten von Fingern	Kleine Bewegungen in 3D
Erkennung mehrerer Arme	Arm- und Hand-Tracking, Design von Interfaces
Kombinierte Berührungserkennung	Kombination von Objektverfolgung und Berührungseignissen

werden Methoden vorgestellt, die eine Lokalisierung, sowie Sturzerkennung auf Böden ermöglicht, und ein verbessertes Verfahren zur Gestenerkennung bei kleiner Sensorzahl. Modellbasierte Fitting-Verfahren ermöglichen es physische Eigenschaften realer Körper über abstrahierte Modelle zu gewinnen, indem diese auf die gewonnen Sensordaten abgebildet werden. Hier werden zwei Methoden für Einkörper, und Mehrkörpermodelle vorgestellt. In beiden Fällen soll die Pose des menschlichen Körpers über intelligenten Möbeln erkannt werden. Ein weiteres Gebiet sind heterogene Sensorsysteme. Hier werden zwei Verfahren vorgestellt. Heterogene Geometrien ermöglichen die parallele Verarbeitung und Messung verschiedener Objekte, während heterogene Technologien genutzt werden, um die Schwächen kapazitiver Sensorik bei der Erkennung von Berührungen zu umgehen. Bildbasierte Verarbeitung kapazitiver Daten nutzt Position der Sensoren, sowie den Messwert, um zunächst ein Bild zu errechnen, das über gängige Verfahren der Bildverarbeitung bearbeitet werden kann, um verschiedene Information über die Objekte im Feld zu generieren. Hier wird ein neues Verfahren vorgestellt, das die Position mehrerer Arme über einer Oberfläche bestimmen kann. Zuletzt werden zwei Verfahren vorgestellt die über kapazitive Sensorik physiologische Parameter bestimmen. Dies sind die Messung der Atemfrequenz über eine Signalanalyse im Frequenzraum, sowie die Erkennung von Schlafphasen basierend auf einer Analyse der Bewegungsmuster.

Um die entwickelten Verfahren zu testen wurden verschiedene Prototypen entwickelt. Die unter diesem Abschnitt befindliche Darstellung zeigt Skizzen dieser Systeme, in welchen die Positionen der Sensorelektroden markiert wurden. CapFloor ist ein kapazitives System zur Innenlokalisierung und Sturzerkennung [BHW12]. Ein Gitter von Drähten kann unter verschiedensten, nicht-leitenden Bodenbelägen angebracht werden. Die Elektronik ist am Rand angebracht, was eine einfache Wartbarkeit und hohe Flexibilität ermöglicht. Das Smart Bed nutzt acht kapazitive Elektroden die am Lattenrost angebracht werden. Durch diesen Aufbau reagieren sie auf Präsenz und Druck, was das Signal verstärkt [BH12]. Es ist in der Lage die Präsenz und Pose von ein oder zwei Personen zu erkennen und ermöglicht über eine Bewegungsanalyse die Erkennung von Schlafphasen [DBM14]. Der Capacitive Chair ist ein Bürostuhl, in welchen acht kapazitive Sensoren integriert wurden. In einem Falle ist dies ein leitfähiger Faden, der in ein bestehendes Mesh-Gewebe eingebracht wurde [BF14]. Der Stuhl ist in der Lage die genaue Position verschiedener Körperteile zu ermitteln und kann über einen Klassifizierer verschiedene Posen erkennen. Zudem findet eine Frequenzanalyse einzelner Sensorsignale statt, um die Atemfrequenz einer Person zu bestimmen. Der Active Armrest ist eine PkV-Armlehne die mit kapazitiver Sensorik erweitert wurde [BNS*14]. Es ist ein heterogenes System aus zwei Sensoren zur Bestimmung der Armpose, sowie ein 6-Elektroden Array, welches Fingergesten erkennt. Hierüber können verschiedene sekundäre Funktionen eines Autos gesteuert werden, wie z.B. ein Multimediasystem. Die MagicBox ist ein kapazitives Gestenerkennungssystem, welches über nur sechs Sensoren operiert [BDK13]. Es registriert die Handposition in einem Abstand bis zu 30 cm. Ein Trainings- und Klassifizierungssystem ermöglicht es eine Vielzahl von Gesten zu unterscheiden. Der



Skizzen aller Prototypen, sowie der Anordnung der Elektroden

CapTap ist ein Wohnzimmertisch, in welchen ein heterogenes kapazitiv-akustisches Gestenerkennungssystem unsichtbar integriert wurde [BZWK*14]. 24 kapazitive Sensoren ermöglichen die Rekonstruktion eines kapazitiven Bildes und Anwendung verschiedener Bildverarbeitungsverfahren. Das akustische System ermöglicht die Erkennung verschiedener Berührungsgesten, die über kapazitive Abstandssensoren sonst nicht erkennbar wären. Zudem werden verschiedene weitere eigene Prototypen, sowie Systeme aus der Literatur erläutert. In Kapitel 5 werden kapazitive Abstandssensoren im Bereich Smart Environments evaluiert. Mit den Erfahrungen aus Literaturanalyse und der Erstellung der verschiedenen Prototypen ist es somit möglich diese Technologie qualifiziert einzuordnen. Zunächst findet ein Vergleich mit anderen populären Sensortechnologien statt, die im Rahmen der verwandten Arbeiten eingeführt wurden. Anschließend werden spezifische Vor- und Nachteile herausgearbeitet. Dies führt im Anschluss zu einer Sammlung von Richtlinien, welche den Designprozess für Anwendungen in Smart Environments unterstützen, insbesondere im Hinblick auf Wahl der Sensortechnologie, sowie Selektion geeigneter Datenverarbeitungsschritte.

Contents

1. Introduction	1
1.1. Motivation	1
1.2. Research Challenges	3
1.3. Contributions	4
1.4. Structure of this work	5
2. Related Work	7
2.1. Electric field sensing	7
2.1.1. Physical properties	7
2.1.2. Proximity sensing versus touch sensing	9
2.1.3. Measuring modes	10
2.1.4. Materials and geometry	11
2.1.5. Data processing	13
2.2. Capacitive proximity sensing applications	19
2.3. Sensor systems in smart environments	27
2.3.1. RGB cameras	28
2.3.2. Infrared cameras	30
2.3.3. Ultrasound sensors	32
2.3.4. Microphone arrays	33
2.3.5. Radiofrequency sensing	35
2.4. Applications in smart environments	37
2.4.1. Indoor localization	39
2.4.2. Gestural interaction	40
2.4.3. Physiological sensing	41
2.4.4. Activity recognition	42
2.4.5. Smart appliances	43
2.4.6. Mobile devices	44
2.4.7. Autonomous systems	45
2.5. Disucssion	46
3. Benchmarking model for sensors in smart environments	47
3.1. Benchmarking	47
3.2. Sensor features	48
3.2.1. Sensor performance characteristics	48
3.2.2. Pervasive metrics	49
3.2.3. Environmental characteristics	49
3.2.4. Discussion and feature matrix	49
3.3. Benchmarking model	51
3.3.1. Feature score and weighting	51

3.3.2.	Modeling	52
3.3.3.	Feature score normalization	53
3.3.4.	Scoring	53
3.4.	Evaluating technology popularity	54
3.4.1.	Scoring	54
3.5.	Results	56
3.5.1.	Central tendency bias	56
3.6.	Applications for capacitive proximity sensors	57
3.6.1.	Benchmark weights	57
3.6.2.	Benchmarking scores for capacitive proximity sensors	60
3.7.	Discussion	61
4.	Use cases for capacitive proximity sensors	63
4.1.	Use cases and associated challenges	63
4.2.	Processing methods	66
4.2.1.	Sparsely distributed sensor arrays	67
4.2.2.	Model-driven fitting methods	70
4.2.3.	Heterogeneous sensor systems	76
4.2.4.	Image-based processing	81
4.2.5.	Physiological signals in frequency- and time-domain	84
4.3.	Application prototypes	89
4.3.1.	CapFloor	89
4.3.2.	Smart Bed	92
4.3.3.	The Capacitive Chair	95
4.3.4.	Active Armrest	101
4.3.5.	Magic Box	104
4.3.6.	CapTap	107
4.4.	Other capacitive prototypes	116
4.5.	Capacitive prototypes from related work	120
4.5.1.	Indoor localization	120
4.5.2.	Smart Appliances	121
4.5.3.	Physiological sensing	121
4.5.4.	Gestural interaction	122
4.6.	Discussion	123
5.	Evaluating capacitive proximity sensors in smart environments	125
5.1.	Comparison to other sensing technologies	125
5.1.1.	Overview of sensing technologies in smart environments	125
5.1.2.	Classification of capacitive proximity sensors	127
5.2.	Benefits	128
5.2.1.	Versatility	128
5.2.2.	Unobtrusiveness	129
5.2.3.	Processing Complexity	130
5.3.	Limitations	131
5.3.1.	Environmental Influence	131
5.3.2.	Physical Range	133
5.3.3.	Object Detection	134

5.4. Guidelines	135
5.4.1. Sensor choice	135
5.4.2. Hardware design	136
5.4.3. Algorithms	137
5.5. Discussion	138
6. Conclusions and Future Work	139
6.1. Future Work	141
A. Evaluation results	145
A.1. Capacitive chair posture recognition	145
A.1.1. Evaluation setup	145
A.1.2. Raw results	145
A.1.3. Postures	146
A.2. Capacitive Chair - working situation recognition	147
A.3. CapTap evaluation	148
A.3.1. Questionnaire	148
A.3.2. Raw results	149
A.4. Active Armrest evaluation results	152
A.4.1. Questionnaire	152
A.5. CapFloor @ EvAAL 2011	153
A.6. Smart Bed sleep phase recognition results	154
B. Publications	155
C. Supervising Activities	159
D. Curriculum Vitae	161
Bibliography	163

List of Figures

1.1. Sketch of electrode placement of all capacitive sensing prototypes created in the scope of this work	2
2.1. Black box setup of a capacitive proximity sensor	8
2.2. Capacitive sensing procedure	9
2.3. Different projected capacitive sensing methods based on distance	9
2.4. Three measurement modes for capacitive proximity sensing [Smi96]	10
2.5. Spatial resolution of different materials at various distances [GPBB*13]	11
2.6. Examples of multilayer layouts for touch screens - grid (a), interlocking diamonds (b) and trade-marked complex patterns (c) [BO10a]	12
2.7. Electrode placement below upholstery (adapted from [GPMB11])	13
2.8. Abstracted sensor data processing pipeline	14
2.9. Example of baseline reset using a threshold rule	15
2.10. Generic pipeline of probability based methods of capacitive proximity sensing	17
2.11. <i>Left:</i> Leon Theremin playing his eponymous electronic musical instrument [Gli00]. <i>Right:</i> The Theremini by Moog Music Inc., released in 2014 [Moo14].	19
2.12. <i>Left:</i> Concept view of passenger seat set to deploy or not deploy airbag. <i>Center:</i> Sensor readings for empty seat and adult person. <i>Right:</i> Sensor readings for front-facing child seat (FFCS) and rear-facing child seat (RFCS). [Smi99]	20
2.13. <i>Left:</i> LaZmouse innards <i>Center:</i> Joshua R. Smith using LaZmouse [Smi99]. <i>Right:</i> Novint Falcon 3D input device [Nov14]	20
2.14. <i>Left:</i> Person interacting with the gesture wall <i>Right:</i> Air drawing results, depth estimation results and associated movements on bottom [SWD*98].	21
2.15. <i>Left:</i> Prototype of CapToolKit [WKBS07]. <i>Right:</i> Thracker prototype and visualized grasping gestures [WHKS06].	22
2.16. <i>Left:</i> Prototype of HandSense and supported grasping types [WB09]. <i>Right:</i> Setup of time domain reflectometry sensing and example of stretchable material [WB11].	23
2.17. <i>Top left:</i> Technical configuration of SmartSkin. <i>Top Right:</i> Bicubic interpolation method to detect the peak of the potential field created by hand proximity. <i>Bottom left:</i> Visualized sensor effect of hand hovering in 10cm distance and touching the surface. <i>Bottom right:</i> Capacitance tag having a specific conductive pattern that can be identified [Rek02].	24
2.18. <i>Top:</i> Capacitive profiles of different gestures on a door knob [SPH12]. <i>Bottom Left:</i> User identification using capacitive fingerprinting [HSP12]. <i>Bottom Right:</i> Botanicus Interacticus prototype and interaction concept [PSLS12].	25
2.19. <i>Top:</i> Measurement of TileTrack resolution with five integrated receivers [VV12]. <i>Bottom Left:</i> Simulation of transmitted field by a single person [VKMV11]. <i>Bottom Right:</i> Using TileTrack system for interaction [VRV10].	26
2.20. <i>Left:</i> GestIC sabrewing prototyping board [Mic13]. <i>Right:</i> Transparent interaction device based on GestIC [LTI*14].	27
2.21. A bayer pattern on a sensor in isometric perspective [Wik06]	29

2.22. <i>Left</i> : Tracking of body masks using cameras [TKA06]. <i>Right</i> : Eigenfaces created from input picture set [PC00].	29
2.23. <i>Left</i> : Eulerian Video Magnification to attenuate the human pulse with original (a) and amplified (b) video sequence [WRS*12]. <i>Right</i> : FoodBoard schematics (top), underside view (second row), original, reconstructed and segmented image (third row) and final system (bottom) [PJS*13].	30
2.24. <i>Top Left</i> : ORL Active Badge [Wei91]. <i>Top Right</i> : Kinect infrared projection [Zha12]. <i>Bottom Left</i> : Kinect Fusion reconstruction [IKH11]. <i>Bottom Right</i> : Kinect kitchen interaction [Pan12]. . .	31
2.25. <i>Top Left</i> : Cricket Indoor Localization hardware [PCB00]. <i>Top Right</i> : Mid-air particle manipulation [OHR14]. <i>Bottom Left</i> : Robotic wheelchair MAid with ultrasound range sensors [PSF01]. <i>Bottom Right</i> : Activity and context recognition using ultrasound sensors [WTT13].	32
2.26. <i>Top Left</i> : Visualization of speaker count in different areas [XLL*13]. <i>Top Right</i> : Directional microphone and conference room for speech source localization [ZFBZ08]. <i>Bottom</i> : TapSense detecting different tap events [HSH11].	34
2.27. <i>Top left</i> ZigBee node. <i>Bottom Left</i> Sensors and targets in larger room [SKOM06]. <i>Top right</i> Photo of radio tomography setup. <i>Bottom right</i> Result attenuated signal [WP10].	36
2.28. <i>Top left</i> WiTrack antennas and signal generator. <i>Bottom Left</i> WiTrack 2D and 3D localization method [AKKM14]. <i>Top right</i> WiSee supported gestures. <i>Bottom right</i> WiSee Doppler profiles of gestures [PGGP13].	37
2.29. <i>Left</i> : Coverage of a room by camera fields of view - AmbiTrack [BDA*13]. <i>Middle</i> : Fiduciary marker photographed by camera phones [MWBS09]. <i>Right</i> : Magnetic coils in background and receiver circuit in foreground [PL13].	39
2.30. <i>Left</i> : FTIR multitouch table [Han05]. <i>Middle</i> : DataGlove hand gesture system [ZLB*86]. <i>Right</i> : XWand interaction device [WS03].	40
2.31. <i>Left</i> : MACH conversation coach system [HCM*13]. <i>Right</i> : Study setup to collect emotion data [KMS*12].	41
2.32. <i>Left</i> : Eye movement tracker by Bulling [BWGT11]. <i>Right</i> : Hon4D - surface normals of different activities [OL13].	42
2.33. <i>Left</i> : Different MediaCup prototype [GBK99]. <i>Right</i> : Stickear system to augment appliances with audio sensing [YNR13].	43
2.34. <i>Left</i> : TiltRacer controlled by accelerometers of mobile phones [VCBE07]. <i>Right</i> : PECo environment control application on a PDA [SN04].	45
2.35. <i>Left</i> : Scenario of the PEIS system [BGS*06]. <i>Right</i> : Robots moving around in a social environment [GKIH09].	46
3.1. Benchmarking process	51
4.1. Piecewise linear hand distance estimation [BH11]	67
4.2. Principle components of a learning by example recognition framework [BDK13]	68
4.3. Wire electrode grid below floor cover attached to sensors on the border	69
4.4. Two potential person locations resulting in same sensor readings (green indicates active electrodes	69
4.5. Shapes of a standing and lying person on top of the CapFloor grid	70
4.6. Object on mattress decreases distance and changes geometry of flexible electrode [BH12]	71
4.7. Cylindrical human body model and various poses on mattress [BH12]	71
4.8. Pressure distribution of a uniform cylinder [BH12]	72
4.9. Calculating centers of pressures and deviation [BH12]	73
4.10. Smartchair skeleton model and associated body parts	74
4.11. Flowchart of the model fitting process of the grouped skeleton parts and performed calculations	75

4.12. Screenshots of the Capacitive Chair application showing different poses of the skeleton model .	76
4.13. Selected set of postures from Global Posture study and own gestures. <i>From top left to bottom right:</i> The strunch, the draw, the smart lean, the take it in, upright, no person (first four taken from [Ste13])	77
4.14. Heterogeneous sensor array for limb detection and finger tracking	78
4.15. Arm model and detection of posture based on distances to two sensors and finger array for resting position (1), hand raised position (2) and arm raised position (3)	78
4.16. Suspended peg knock detection system	79
4.17. 64 sample FFTs and photo for a knock event (A), a finger tap (B), a finger swipe (C) and a hand swipe (D)	80
4.18. Pixel array mapped from sensor values	82
4.19. Effect of different upscaling methods on shape, (A) nearest neighbor, (B) bicubic, (C) bilinear, (D) Lanczos4 - shown as thresholded binary images (pixel intensity > 30)	82
4.20. <i>Left:</i> Capacitive image of two arms, detected palm position and arm orientation. <i>Right:</i> Overhead camera picture of the scene overlaid with live arm and palm reconstruction for two arms	83
4.21. Chest movement when breathing in and out	85
4.22. Sinusoidal curve caused by chest movement above a capacitive sensor placed close to the chest .	86
4.23. Example of human sleep phases throughout the night	86
4.24. CapFloor sketch - grid layout of electrodes is placed below a floor layer with sensors attached on the sides	90
4.25. Floor mats with integrated CapFloor system used at the EvAAL 2011 competition [BHW12] . .	91
4.26. Smart Bed sketch - flexible plate electrode are attached on spring board	92
4.27. Data processing components [BH12]	93
4.28. Electrodes and sensors hidden below mattress of Smart Bed [BH12]	93
4.29. Sleep movement data over three hours in one night [DBM14]	94
4.30. Smart office chair sketch - eight electrodes three in backrest, three on seat and two in armrests .	95
4.31. Capacitive Chair electrode positions	96
4.32. Detail view of conductive thread electrode	97
4.33. Screenshot of the Capacitive Chair application showing the fitted 3D model on the left and detected posture on the right	98
4.34. Screenshot of the Capacitive Chair application showing the sensor values on the left and FFT analysis of various sensors on the right	99
4.35. Work Activity aggregation over a single work day (mock-up)	100
4.36. Example chart of work activity data collected	100
4.37. Active armrest sketch - six electrodes for finger gesture detection in front, two for arm detection in back	101
4.38. Active Armrest prototype, left - outside view, right - detail view of electronics [BNS*14].	101
4.39. Postures of limbs on armrest - resting position (left), arm raised position (middle), hand raised position (right) [BNS*14].	102
4.40. Active Armrest demo software, left - finger tracker, right - OSM based navigation application [BNS*14]	102
4.41. Confusion matrices of recognized gestures for touch interaction (left) and free-air interaction (right)	103
4.42. MagicBox sketch - six electrodes uniformly distributed below surface	104
4.43. MagicBox conceptual rendering (left) and detail view of electronics (right) [BH11]	104
4.44. MagicBox demonstration application - 3D object viewer (left) and image viewer (right) [BH11]	105
4.45. Gesture overview module (left) and gesture recorder (right) [BDK13]	106
4.46. CapTap sketch - 24 electrodes placed under table surface and a single detector for touch events .	107

4.47. Detail views of the prototype system: left - electrodes and sensors, right - audio interface and contact microphone	108
4.48. Abstract view below the surface of the prototype including capacitive sensing electrodes and touch detection microphone	108
4.49. Views of final prototype, complete view (left), top view with markers for touch evaluation (right)	109
4.50. Multiple interaction layers	109
4.51. <i>Top</i> : Progression from far layer to middle layer image thumbnail sizes. <i>Bottom</i> : Different zoom levels controlled by arm orientation.	110
4.52. <i>Left</i> : Recognized disturbing arm. <i>Right</i> : Arm orientation when leaning in.	111
4.53. Tracks generated by Kalman filtered palm position, a sine wave (A), a rectangle (B), a circle (C), and a diagonal swipe (D)	112
4.54. Finger tap (blue), knuckle knock (green), finger swipe (purple), hand swipe (orange) and stomping (red) spots relative to tabletop.	113
4.55. Interaction speed evaluation. Different types of boxes for near layer (N), knock (K), far layer (F), and disturber (X).	114
4.56. CapDisp sketch - TV on a stand equipped with capacitive sensors hidden below a plastic cover .	116
4.57. <i>Top left</i> : Person lying on the couch. <i>Top right</i> : Resulting sensor value visualization. <i>Bottom left</i> : Rendering of recongized posture. <i>Bottom right</i> : Position of electrodes within the couch.	117
4.58. <i>Top left</i> : Rendering of Honeyfish device and hands. <i>Top right</i> : Image of developed GUI and pointer. <i>Bottom left</i> : Swiss cheese algorithm predicting objects in interaction space. <i>Bottom right</i> : Evaluation of Honeyfish at a student fair.	118
4.59. <i>Top left</i> : Concept image of GestDisp in a car with outlined electrodes. <i>Top right</i> : GestDisp prototype installed in front of a monitor. <i>Bottom left</i> : Screenshot of demonstration application music player. <i>Bottom right</i> : Association of gestures to functions in the demonstration application.	119
4.60. <i>Left</i> : Spread spectrum capacitive sensor detecting respiration 30cm (green) and 90cm (blue) from chest. <i>Right</i> : Same sensor detecting person walking by at 100cm (left), 70cm (middle) and 35cm (right) [Mac04].	122
5.1. Sketch of CapTap prototype showing thickness of tabletop	130
5.2. Reduced angular resolution on smaller, distant objects	133
5.3. Same response to differently sized objects (left), different response to varying materials (right) .	134
5.4. Basic system design process for capacitive proximity sensing systems and associated aspects in the single steps	135
A.1. <i>Top left</i> : Upright posture. <i>Top right</i> : Hunched posture. <i>Bottom left</i> : Slouched posture. <i>Bottom right</i> : Disturber posture	146
A.2. Work activity day 1	147
A.3. Work activity day 2	147
A.4. Work activity day 3	148
A.5. Mean values and standard deviations for the Likert items of Active Armrest questionnaire	152
A.6. Recognition rate of CapFloor on selected areas of interest	153
A.7. Recognition of sleep phases over three nights	154

List of Tables

2.1. Baseline calibrations terms and methods	14
2.2. Overview of high-level processing methods for capacitive proximity sensors	16
2.3. Feature recognition methods	18
2.4. Sensors for smart environments [CD07]	28
2.5. Sensing technologies and measured properties	28
2.6. Application domains and relevant works	38
3.1. Feature matrix denoting capabilities required for a certain rating	50
3.2. The importance weighting of different applications, based on the features.	54
3.3. Feature rating of the different sensor technologies	55
3.4. Regular and normalized benchmark score matrix of different applications and technologies	55
3.5. Search result frequency given specific applications, sensor technologies and synonyms for ACM Digital Library (DL) and Google Scholar (GS)	55
3.6. Central tendency bias correction for different exponents a	57
3.7. Importance weights of applications for capacitive proximity sensors	58
3.8. Feature weights for capacitive proximity sensors	59
3.9. Benchmarking scores for capacitive proximity sensors in different applications	60
4.1. Application domains and derived implemented use cases for capacitive proximity sensing	63
4.2. Challenges associated to the different use cases for capacitive proximity sensors	66
4.3. Overview of developed capacitive proximity sensing prototypes	89
4.4. Results of touch detection for single and double taps (SFT, DFT), knocks (SKK, DKK), finger swipe (FS), hand swipe (HS) and stomp (STO). Noted are the overall samples, errors, no event errors, wrong classification errors and the percentage of correct classification.	112
4.5. Results for interaction time in the different runs of the interaction speed test	114
4.6. Measuring layout and data processing of different prototypes from related works	120
5.1. Qualitative comparison between capacitive proximity sensors and other sensor technologies	126
5.2. Overview of capacitive proximity sensing benefits	129
5.3. Overview of capacitive proximity sensing limitations	132
A.1. Percentage of correctly classified postures using manually set classifier	145
A.2. CapTap evaluation raw results	150
A.3. CapTap questionnaire results	151
A.4. Best case scores of CapFloor @ EvAAL 2011	153

1. Introduction

Smart environments are comprised of numerous sensing and computing devices. They support a number of users in this environment in performing tasks. Driven by advances in computing power, miniaturization of sensors and processing methods, novel devices that have a plethora of functionality are introduced into our everyday lives. As a scientific field it has been thriving in the last decades. It combines knowledge from disciplines including computer science, engineering, and product design. Systems are created that are integrated into the environment, have high usability, and provide information and services. Perhaps the most cited example of this trend is the rise of the smartphone, from a professional business tool to a consumer device, being sold hundreds of millions times each year. Using integrated sensors and communication facilities, it provides services aware of location, schedule, contacts, or preferences. Application include navigation, event planning, augmented reality, or multimedia playback. Another example is increasingly connected homes that are aware of energy usage, lighting levels, temperature and the status of critical devices. They can be controlled by the user with a single device, e.g. a smartphone, or autonomously using a set of specified rules.

Sensing is a common aspect of all smart environments and smart devices. The sensing properties includes environmental parameters, system state, and the activities of the different users. Cameras, accelerometers, GPS, or acoustic sensors are common examples of sensing devices. Capacitive sensors are a category of sensors that use electric fields to sense the presence and certain properties of the human body. The most common variety is sensing the presence of fingers on touch screens, which is already present in billions of devices. However, there is another variety, the capacitive proximity sensor. It is able to detect the presence of the human body or other conductive objects over a distance, providing applications in smart environments. These sensors can be unobtrusively integrating the sensors into different materials, environments and appliances. When creating an application for smart environments, choosing the right sensors is a critical decisions that has to be taken early in the design process. So far this process has been mostly supported by looking at previous applications and prototypes.

In this work I present a benchmarking model that supports this decision process in the domain of smart environments. Relevant application areas for capacitive proximity sensors can be identified in this process. For each area there are different challenges that have to be considered, including the details of choosing hardware layout and suitable algorithms. I present a collection of improved and novel methods that process data generated by capacitive proximity sensors. To evaluate those methods, several prototypes have been created and tested for performance and usability. Figure 1.1 shows sketches of the different prototypes and the placement of the electrodes attached to the capacitive proximity sensors. Based on these evaluations and the knowledge generated in the design process, I discuss the benefits and limitations of the technology and classify it with regards to competing technologies. Finally, I present a set of guidelines that can aid parties in designing smart environment applications using capacitive proximity sensors.

1.1. Motivation

In the last decade the way we interact with computing machines has changed profoundly. Today more than one billion people operate a smartphone, enabling ubiquitous access to communication tools, processing power and

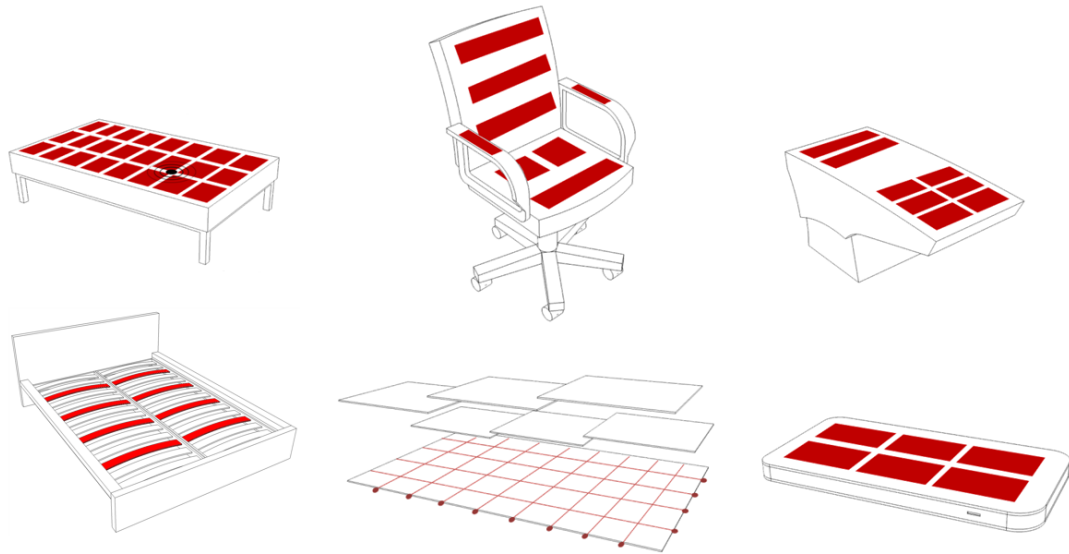


Figure 1.1.: Sketch of electrode placement of all capacitive sensing prototypes created in the scope of this work

information. The vision of Ubiquitous Computing as proposed by Mark Weiser in the early 90s is moving closer to reality [Wei91]. The required technologies of

"cheap, low-power computers that include equally convenient displays, a network that ties them all together, and software systems implementing ubiquitous applications"

are now existing in the form of smartphones and tablets that are connected to the internet, using high-speed connections such as LTE (Long Term Evolution, also known as 4G - a high speed mobile communication protocol), and web-based services such as Google Now, that combine numerous data sources to provide personalized services.

The vision and underlying ideas Ubiquitous Computing have been expanded in the last years, leading to the research areas of Pervasive Computing and Ambient Intelligence. The concept has been expanded to not only consider directly manipulated devices, but also to include devices that determine the current situation and can react based on it. This context-aware computing proposes

"systems that examine and react to an individual's changing context. Such systems can promote and mediate people's interactions with devices, computers, and other people" [SAW94]

Different forms of context can be distinguished, ranging from location and the actual system state, to different activities or even the current mood of the user. In order to acquire this context, the input-and-output based systems originally proposed by Weiser are augmented by an ensemble of devices that are very small (dust), coordinate in massive numbers (clay), or are flexible, unobtrusive extensions to everyday objects (fabric) [Pos11]. These devices can be invisibly integrated into our everyday environment and provide sensing capabilities. Examples of these devices are microelectromechanical systems (MEMS) or mechanically flexible electronics, such as OLED screens. The number of computation and sensing devices that we carry with us is growing. Yet we want the technology to further disappear, allowing us to focus on the application instead of the underlying technology.

The science fiction author Arthur C. Clarke proposed three laws of prediction, the third of which is

"Any sufficiently advanced technology is indistinguishable from magic." [Cla62]

Capacitive proximity sensing measures the influence of the human body (or conductive objects in general) on an electric field. While this technology is not magic per se, a peculiarity of electricity is that humans, as opposed to some animals, have no specific sensing organ for this property. Thus we remain unaware of the electric field, unless the field strength is high enough to cause electric shocks. Consequently, when interacting with capacitive sensors humans have no awareness of the sensed property. However, the measurement can be exposed to another sense. This could be haptic feedback on touch screens or visual feedback on free air interaction systems. Touch screens are the most ubiquitous application of capacitive technology. It is applied in all recent smartphones and tablets, thus being used by a large number of the world's population every day. However, they are typically tuned to only register touch, with some varieties being able to detect fingers over a small distance. This variety is called proximity sensing. It enables numerous other applications for this technology, ranging from industrial fluid level and material detection, to presence detection in cars. A particularly interesting domain for this sensing technology is smart environments. As previously mentioned, they provide services based on unobtrusively acquired information about actors in this environment. In this regard they can be considered fabric - unobtrusive extensions to everyday objects. Capacitive proximity sensors have been primarily used for human-computer interaction (HCI) applications. Examples include a mouse tracking the proximity of the heel of the hand, and a monitor that is able to track gestures performed in front of it. Another application is smart appliances, such as an object detecting car seat or localization systems.

In smart environments there are different sensing technologies that provide similar detection capabilities. Cameras and accelerometers can both distinguish simple activities, such as standing, walking and lying. The difference in this application is that cameras are placed in the environment, while accelerometers have to be carried by the user. The preferred solution depends on the specific application. It is often difficult to decide which specific sensor technology to use in a specific system. Commonly, one refers to existing work and best practice, building on previously generated knowledge. Prior to this work, there is no formal model that would allow designers to quickly evaluate different sensor technologies for different applications. Taking into account a set of features required for a specific application domain the model can be an important decision support tool in the early stages of system development. As it was stated by Cook and Das [CD07]:

"Finally, a useful goal for the smart environment research community is to define evaluation mechanisms. While performance measures can be defined for each technology within the architecture hierarchy [...], performance measures for entire smart environments still need to be established. This can form the basis of comparative assessments and identify areas that need further investigation."

The model can also be used to identify specific applications for a single sensor technology, such as verifying current use cases for capacitive proximity sensors or develop new ones. This allows a classification of the technology with regards to competing sensor systems. However, identifying a suitable sensor is just the first step in designing a smart environment prototype or product. After this decision a designer has to determine specific challenges for his use case, select suitable methods for hardware layout, and algorithms for data processing. In this design step it is helpful to have a set of methods, examples and guidelines, leading to improved rapid prototyping for researchers and shorter time-to-market for product developers. These guidelines should be the result of literature review and validating prototypes for performance and usability.

1.2. Research Challenges

There are numerous influential works that give an overview of technologies and applications in smart environments. Cook and Das identify common technologies, frameworks and applications in this domain and give an overview of ongoing research [CD07]. Poslad specifies a more detailed taxonomy of device classes, provides concepts for interaction between humans and environments and gives an overview of intelligent systems [Pos11].

Another category of previous work details the different sensing technologies that are supporting various applications and provides an overview of limitations and benefits. However, no existing work provides a benchmark that maps sensor characteristics to applications in smart environments. An intermediate step between evaluating entire environments and low-level technologies is an application-specific benchmarking of systems. Benchmarking allows us to quantify the performance of a process or item and allows a comparison to similar entities. It is common to benchmark different technologies according to their features. I therefore extend technology-driven benchmarks by adding an application-specific feature weighting. This approach is introduced in Chapter 3. It allows for mapping of the same set of features to different applications that have similar requirements. These applications can be realized by multiple technologies. It is verified by benchmarking typical sensor technologies with regards to several example applications in the domain of smart environments.

The selection of application scenarios for capacitive proximity sensors is mostly driven by previous work, most notably by research groups from MIT [ZSP*95], Disney research [SPH12] and the Munich University of Technology [WHKS06]. In each case the authors extend the methods or modify existing use cases to another domain. The benchmarking model presented in this thesis extends these methods by providing a formal method that allows the verification of existing use cases, enables a comparison with other sensing technologies, and presents the possibility to find novel applications for capacitive proximity sensors. Chapter 3.6 outlines the benchmarking model and concludes with a collection of four relevant application domains that can be realized using capacitive proximity sensors.

For each of the identified application domains there are distinct challenges. Each challenge requires selecting suitable processing methods and a functional hardware layout that go beyond what is presented in the related works. Accordingly, there are several areas that can be improved using novel or adapted methods. For example, previous systems often rely on uniform sensor arrays [Smi96] or require a large number of sensors [Rek02]. I propose improvements to five different areas in Chapter 4.2. These are object tracking using a restricted sensor count, model-driven approaches for object fitting, heterogeneous sensor systems, image-based processing of capacitive sensor data, and physiological sensing. I have created different prototypes that use the presented methods and evaluated them for usability and performance, as detailed in Chapter 4.

While there are numerous applications based on capacitive proximity sensors, there has been no distinct classification within smart environments. My classification includes a comparison to other sensor technologies, and identifies benefits and limitations of capacitive proximity sensors. Based on knowledge generated from creating and evaluating the different prototypes I am able to classify capacitive proximity sensors in comparison to other sensing technologies in smart environments. This is shown in Chapter 5. This culminates in a set of guidelines for parties interested in developing smart environment systems based on capacitive proximity sensors.

1.3. Contributions

In the following section I briefly list the methodological and practical. They are separated into four different groups: the benchmarking model, the introduction of new or improved processing methods, the creation and evaluation of proof-of-concept prototypes, and the classification of capacitive proximity sensors in smart environments.

1. *Introduction of a generic and formal benchmarking model for sensor systems in smart environments*

The benchmarking model can verify use cases for any given sensor technology in smart environments. For this purpose, I identify the most relevant sensor features for the scoring process and create a feature matrix that links features to application importance rating. Additionally, the benchmarking score calculation is

specified, including normalization and compensation for central-tendency bias. The method is applied to capacitive proximity sensors to find and verify use cases.

2. *New and improved processing methods for capacitive proximity sensors*

For sparse sensor distributions I developed a new method allowing indoor localization and fall detection, as well as new approaches for hand tracking. I propose two novel methods for model-driven processing that detect occupation and pose with both single-body models and multi-body models. Looking at heterogeneous sensor system, I investigate methods to handle non-uniform arrays and provide an integration of capacitive proximity sensors and acoustic systems. I introduce a new method for image-based processing of capacitive sensor data from uniform arrays that tracks multiple objects in three dimensions. Finally, two methods for tracking physiological activities are introduced that operate in time- and frequency-domain. They can be used to detect the sleep phase or respiratory rate.

3. *Proof-of-concept and evaluation of processing methods using a variety of prototypes*

The *MagicBox* enables expressive single-hand gestural interaction with sparse sensor distribution and machine learning gesture classification. *CapFloor* proposes a novel layout for floor-based capacitive indoor localization systems, which allows fall detection, using unobtrusive installation and easy maintenance. The *SmartBed* uses a model-based approach for detecting the pose of one or two persons, a motion analysis to detect sleep phases and frequency analysis to measure the breathing rate. The *Capacitive Chair* is a piece of smart furniture that detects the presence and identity of a user, tracks different postures, and measures breathing rate. The *Active Armrest* uses a heterogeneous sensor layout to enable different forms of interaction in automotive environments. The *CapTap* combines capacitive sensors and microphones in a table-based interaction device, enabling multi-hand gesture recognition in three dimensions using an interaction pattern of multiple layers.

4. *Classification of capacitive proximity sensors in smart environments*

I compare capacitive proximity sensors to other sensor technologies that are commonly used in smart environments. I will establish benefits and limitations of capacitive proximity sensors in comparison to competing sensing systems. Finally, I present a set of guidelines that support designers interested in creating smart environment applications based on capacitive proximity sensors.

1.4. Structure of this work

Chapter 2, *Related Work* specifies the relevant literature. It is grouped into four categories. The first section gives a background on electric field sensing, including relevant historical work and the physical properties. Different sensing categories are outlined, followed by considerations of electrode layouts and data processing methods. The second category of related works discusses applications of capacitive proximity sensors that were created in the last decades, ranging from MIT research in the early 90s, to novel touch classifiers based on different sensing methods. The third category consists of competing technologies that are used in the benchmarking. Finally, I give an overview of existing work, collecting and grouping applications in smart environments. This allows me to identify candidate scenarios for capacitive proximity sensors.

Chapter 3, *Benchmarking model for sensors in smart environments*, introduces the application-specific benchmarking model. The first part of this chapter discusses the sensor features relevant for application in smart environments. The suitable features are divided into three different categories and I discuss the rationale for inclusion or omission in the model. The next part describes the benchmarking model. An application-based feature weighting is introduced, leading to the derivation of the model itself, including the required calculation of an overall rating and a feature score normalization. After that I use the model to score different examples and

validate them using a frequency analysis of search results from two scientific publication databases. Afterwards, the model is used to identify suitable applications for capacitive proximity sensors in smart environments.

Chapter 4, *Use cases for capacitive proximity sensors*, outlines the previously identified use cases. First, the associated challenges for design and processing are identified. Afterwards, different processing methods for capacitive proximity sensors are presented that tackle the specific challenges. This includes methods for sparsely distributed sensor arrays, model-based data fitting, heterogeneous sensor systems, image-based processing and physiological signal processing. Six different prototypes that implement one or more of the processing methods are presented and evaluated: MagicBox, CapFloor, Capacitive Chair, Active Armrest, SmartBed and CapTap. Each of the prototypes has been evaluated for performance and usability. Additionally, three other prototypes are discussed briefly.

Chapter 5, *Evaluating capacitive proximity sensors in smart environments*, discusses and evaluates the results, based on the knowledge gathered in designing, building and testing the prototypes. This chapter has four parts. First, capacitive proximity sensors are compared to the sensor classes introduced in the related works. Subsequently, limitations and benefits of the technology are discussed and linked to sensor features and applications. The chapter concludes with a set of guidelines that help parties interested in creating applications using capacitive proximity sensors.

Chapter 6, *Conclusions and Future Work*, briefly recapitulates the dissertation and introduces potential future areas of research.

There are four appendices. Appendix A includes raw results and additional material from the evaluations performed with the different prototypes. Appendix B lists publications and talks. Appendix C lists Master and Bachelor theses that were supervised or co-supervised. Appendix D contains a short CV.

2. Related Work

In this chapter I will introduce the most relevant literature for this work, including a detailed introduction to specific topics. The aim of this chapter is to provide a basis for both, the benchmarking model that is developed in Chapter 3, and the capacitive proximity sensing methods described in Chapter 4. The related works are distinguished into four parts. At first, I will give a general introduction to electric field sensing, including a discussion on different properties, physical background, the influence of materials and geometry, as well as different data processing methods. Following that, I will present relevant applications that use capacitive proximity sensing, ranging from historical works to recent systems. In the next section, various sensing technologies are introduced that are used in typical smart environment applications. Finally, I will identify and group different applications in smart environments, providing a basis for the applications covered by the benchmarking model.

2.1. Electric field sensing

Electricity, as a force has been known for millennia. However, it has not bloomed until the 18th century, when it was turned from a curiosity into a science over the course of the next 200 years. In 1858 famous chemist and physicist Michael Faraday noted:

Electricity is often called wonderful, beautiful; but it is so only in common with the other forces of nature. The beauty of electricity or of any other force is not that the power is mysterious, and unexpected, touching every sense at unawares in turn, but that it is under law, and that the taught intellect can even now govern it largely. [Jon70]

One of those laws is that different electric charges apply either a repelling or attracting force to each other. For any point in space these forces have a distinct direction and magnitude. The resulting collection of force vectors is called the electric field. Conductive objects that are present in this area modify the properties of the field. Electric field sensing enables measuring field properties at a certain point in space. Using continuous monitoring it is possible to gather information about conductive objects passing through the field by associating measured disturbances to properties of the object. It is possible to gather a multitude of different information about a project. In this section I will give an overview of the physical background, different measurement modes and how to process data acquired by digital sensors. This section is extended from the version presented in [BWK15].

2.1.1. Physical properties

A complete overview of the electrostatic principles of capacitive proximity sensing can be found in the book by Baxter [Bax96], Chapters 2 and 6. In the following paragraphs I will give a very brief introduction to this topic. This is based on the related section in [BWK15]. The basic setup of a typically used sensor is shown in Figure 2.1. The proximity capacitance C_x can be determined using a combination of serial and parallel circuits

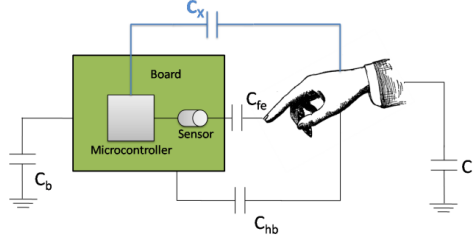


Figure 2.1.: Black box setup of a capacitive proximity sensor

of capacitors, resulting in the following equation:

$$C_x = \left(\left(C_{hb} + \frac{C_h C_b}{C_h + C_b} \right)^{-1} \frac{1}{C_{fe}} \right)^{-1} \quad (2.1)$$

Additionally there is parasitic capacitance, i.e. disturbing capacitance values within the system. It can be caused by a variety of different sources. These include:

- Capacitance of the sensing electrode
- Capacitance between sensing electrode and ground plane
- Intercapacitance between neighboring traces on the board

Looking at a typical system, the combined parasitic capacitances C_{par} amounts to values approximately between $10pF$ and $300pF$ and is therefore considerably larger than the value of the proximity capacitance C_x , being between $0.1pF$ and $10pF$. The total capacitance sensed is the sum of parasitic and proximity components.

$$C_S = C_x + C_{par} \quad (2.2)$$

While this parasitic capacitance can be orders of magnitudes higher than the capacitance that is to be measured, it is typically static in a given system. There are some disturbing factors and changes over time that have to be accounted for. Those will be further discussed in the processing sections. This static value can be determined periodically and filtered out of the resulting value, allowing to get a suitable measurement. The main property of electric field sensing is to measure the distance of objects from the sensing electrode. Thus it is necessary to discuss how the capacitance of common objects approaching the sensor can be estimated. Any object exhibits capacitance in respect to infinity. Surveying simple geometric shapes this capacitance is analytically determinable, e.g.:

$$C = 8\epsilon_0 r_{Disk} \quad (2.3)$$

$$C = 4\pi\epsilon_0 r_{Sphere} \quad (2.4)$$

ϵ_0 is the vacuum permittivity and r the respective radius. This free space capacitance is increasing as soon as another object is approaching, caused by the capacitance of this second object. This results in mutual capacitance, i.e. the resulting capacitive properties between a sending and a receiving object that are close to each other. Looking at generic formulas to determine the capacitance between parallel plates, this behavior can be described analytically.

$$C = \frac{Q}{V} \quad C = \epsilon_0 \epsilon_r \frac{A}{d} \quad (2.5)$$

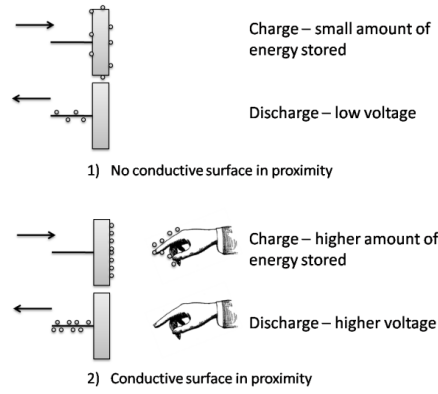


Figure 2.2.: Capacitive sensing procedure

The capacitance is directly proportional to the plate area A and inversely proportional to the distance d between the plates, with ϵ_r being the relative static permittivity of the dielectric between the plates. The sensor electronics are grounded with the body acting as ground itself. The sensor plate is continuously charged using a constant voltage V . A higher capacitance allows the system to hold a larger charge. If the system is connected to the ground, the sensor capacitor is discharged through a resistor. The resulting voltage is depending on the available charge Q , as shown in the equation above. Furthermore, the required time to discharge the capacitor is increased. This effect can be used to measure the capacitance using different techniques for counting the discharge time. This process is symbolized in Figure 2.2. Other varieties use a field between two electrodes. In this case the energy stored is reduced by grounded objects. I will introduce the differences between those sensing modes later in this section.

2.1.2. Proximity sensing versus touch sensing

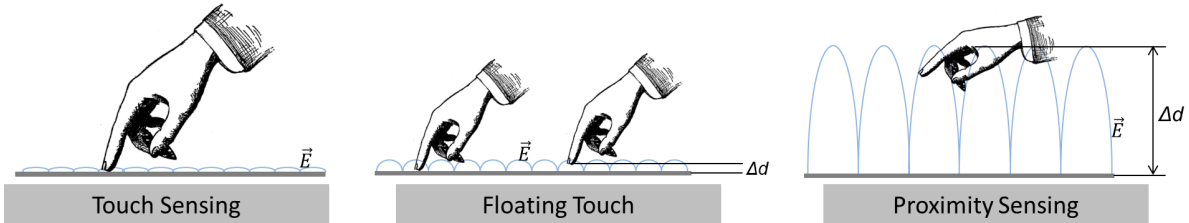


Figure 2.3.: Different projected capacitive sensing methods based on distance

The most ubiquitous use case for capacitive sensing technology is touch screens. As the trend went from pen-controlled mobile systems to finger controlled devices with the first iPhone in 2007, projected capacitance touch is the most prevalent technology for touch screens. It uses various layers of transparent electrodes or very thin wires to measure the mutual capacitance as objects enter the detection area [BO10b]. The commercially available devices have gained additional abilities over the last few years, leading to the development of "floating touch" systems that are able to track fingers in gloves, or fingers that are hovering above the surface [Cyp12, Nok12].

Applications are the usage of mobile devices in cold outdoor temperatures or additional navigation features based on the hovering fingers. In consequence it is possible to distinguish the three different projected capacitive sensing methods shown in Figure 2.3:

- Touch sensing - densely distributed sensors are tuned to project a weak electric field, in order to detect one or more objects touching the interactive surface. The sensors have to be close to the surface.
- Floating touch - densely distributed high-sensitivity sensors are able to detect both touches and very near objects ($< 2cm$) to enable usage using protective gear or additional navigation feature. The sensors have to be close to the surface.
- Proximity sensing - sparsely distributed sensors create a stronger electric field that propagates into space, in order to detect larger objects, such as hands, that are in proximity of the interactive surface. Achievable distances exceed 30 centimeters and the sensors may be applied below thick non-conductive material.

2.1.3. Measuring modes

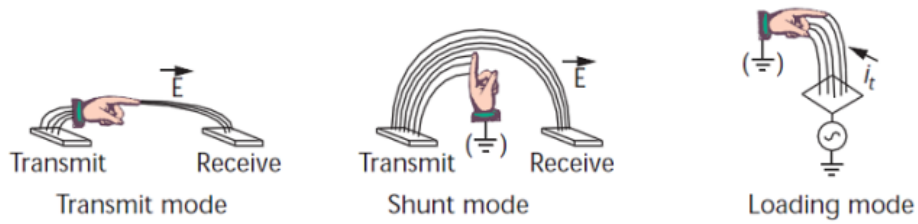


Figure 2.4.: Three measurement modes for capacitive proximity sensing [Smi96]

A classic work in the field of capacitive proximity sensing that will be referenced occasionally in this work is the doctoral dissertation "Electric Field Imaging" by Joshua Smith [Smi99] that reports on the research performed at the MIT Media Lab in the 1990s. One contribution was the introduction of different measurement modes that can be distinguished in capacitive sensing [Smi96]. They are shown in Figure 2.4.

Transmit mode is using a transmitting electrode that is coupled to a conductive object. In case of human-machine interaction, typically the human body. The properties of an electric field generated with respect to a receiving electrode will therefore be dependent on the distance of this body, thus extending the achievable range from the transmitting electrode.

Shunt mode similarly uses both a receiving and transmitting electrode generating a static field. However, there is no body coupled and any conductive object will ground the field, thus reducing the energy stored, which is measured. This setup is able to work with various transmitters on a single receiver, enabling a higher amount of virtual sensors using limited hardware.

Loading mode relies on an oscillating field that is induced on a single electrode. The capacitance is measured relative to the environment. Any approaching grounded object results in an increased system capacitance that is measured periodically.

2.1.4. Materials and geometry

Two major factors that have to be considered when designing an application based on capacitive sensors are the materials and geometry of the electrodes performing the measurements. The material of the electrode should be picked according to the desired application. If the interaction device has a flexible surface, conductive thread could be used, if it is solid and opaque, the application of solid metal electrodes is viable. Additionally, there are other options for transparent materials.

While we traditionally associate solid metals to antennas and electrodes this view can no longer be upheld. Transparent conductive layers have been in use for decades now, e.g. in car windows or solar technology. They typically rely on metal oxide layers, polymer layers or in recent years carbon nanotubes [MPLK05].

In the scope of his Master's thesis Yannick Berghöfer evaluated different types of electrode materials in terms of their spatial resolution at different distances between object and electrode. The results were included in the following paper of our group [GPBB*13]. The focus was to establish how the different materials perform at larger distances. The benchmarked materials included both ITO and PEDOT:PSS. The first is a thin layer of indium-titanium-oxide, a highly conductive metal layer that possesses good optical properties. PEDOT:PSS is a conductive polymer that has a lower conductivity and slightly less appealing optical properties. In conclusion, it could be established that while copper has still the most favorable properties, at least ITO can be considered a suitable alternative in applications that require optical transparency. An overview of the achievable spatial resolutions of the different materials and electrode sizes is given in Figure 2.5. The spatial resolution is a measure that describes the expected precision of the measurement based on taking a time series of multiple samples and determining the mean distance and standard deviation [GPBB*13].

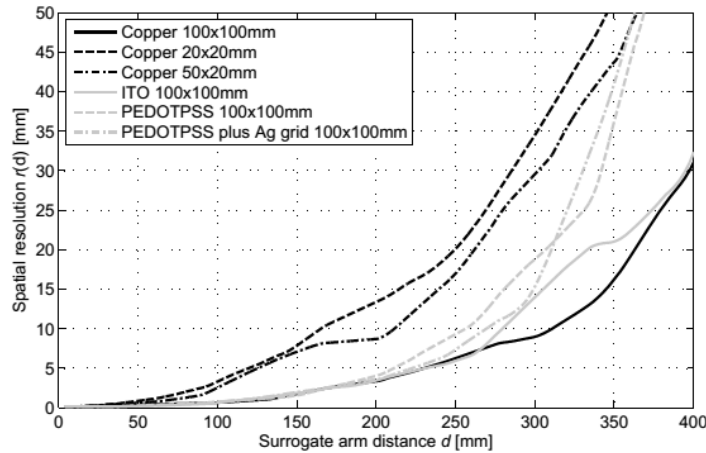


Figure 2.5.: Spatial resolution of different materials at various distances [GPBB*13]

The most common technology for usage in displays is projected capacitive touch that uses a multi-layer design of insulated ITO electrodes that are able to detect the movement of several objects close to the surface [BO10b]. However, they are typically tuned to allow operation within a small distance of 1cm or less. Another area that is strongly influenced by the intended application is the geometry, whereas the electrode is considered the part of the electronics directly attached to the measurement circuit. This may range from simple straight wires or plate electrodes to complex optimized multidimensional structures specifically designed for a single task. Even though it is aimed at touch or near-proximity sensing I will give a short overview of multi-layer designs for touch screens

that have been reviewed by Barrett and Omote [BO10a]. They are designed to measure mutual capacitance. If a sensible excitation and measuring process is used, multiple nearby objects may be reliably detected.

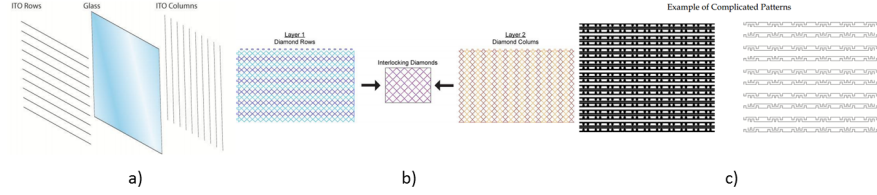


Figure 2.6.: Examples of multilayer layouts for touch screens - grid (a), interlocking diamonds (b) and trademarked complex patterns (c) [BO10a]

A simple example is two layers of perpendicular straight line electrodes - used by the first iPhone (Figure 2.6 - a). Another example uses an interlocking diamond shape [DL01] to create a good spatial coverage (Figure 2.6 - b). Finally, there are numerous other complex patterns that are often trademarked by the companies that have developed the respective controller. One example is given in (Figure 2.6 - c).

Capacitive proximity sensing applications are typically less concerned about intricate designs, but instead use varying electrode sizes and placement over a larger area. As previously mentioned the purpose of capacitive proximity sensing is the detection of objects and their properties. There are numerous factors that can influence the geometrical layout, but they can be abstracted into the following categories:

- Number of objects
- Object size
- Desired spatial resolution

Going back to our example of touch screens, there are small objects, a higher number of those (usually up to 10) and require a high spatial resolution to select small items on the screen. The result is a fine multilayer grid, using mutual capacitance to simplify multi-object recognition, fine electrode spacing to achieve a high spatial resolution and thin wires or transparent electrodes to guarantee good optical properties. A similar rationale can be applied to other applications. Taking the smart couch created by Tobias Große-Puppenthal, Alexander Marinc and myself, the aim is to detect the presence and posture of one or more persons on a couch [GPMB11]. This necessitates detecting large body parts such as head, torso or limbs. There is no fine-grained spatial resolution required, allowing a reduction the number of sensors and it was assumed that a maximum of two persons are on the couch. Furthermore the electrodes are placed below the upholstery, thus requiring a reasonable detection distance.

The resulting electrode placement can be seen in Figure 2.7. The layout was designed under the additional constriction of using a single sensor kit, supporting up to eight electrodes. Regarding placement, it is most important to distinguish two persons and different sitting positions, thus four electrodes are placed below the sitting

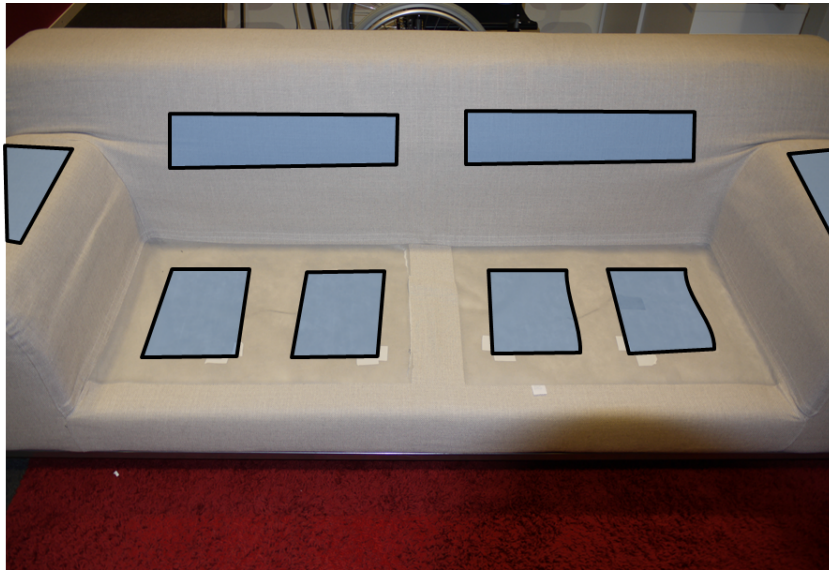


Figure 2.7.: Electrode placement below upholstery (adapted from [GPMB11])

area. In the back there are two electrodes spread over the entire width to determine the presence of the upper body close to the backrest. The electrodes in the armrests detect the head of a lying person. In consequence, this setup is suitable for detecting multiple sitting persons, infer information about their sitting position and recognize lying persons. Regarding those postures it showed good results in the prototype's evaluation [GPMB11].

A third and final example for the rationale of electrode placement is the TileTrack system by Valtonen et al, a capacitive person tracking system using floor tiles [VMV09]. It is a transmit mode system that has the transmitting electrodes placed below the floor tiles. The receiving electrodes are placed in the walls of the area. The main goal of the system is the tracking of persons on the surface. Thus the floor area should be mostly covered by electrodes to establish a good transmission link to the bodies. The receiving electrodes should be able to pick up all signals generated by the body. Valtonen et al. picked wire or plate electrodes that went from floor level to a height of 190cm that covers most typical body sizes. While the system has some shortcomings with regard to applicability in larger rooms, the design rationale is appropriate for narrow rooms or when only movement close to walls has to be detected. In those instances they achieved a good precision in their evaluation. Another proposed extension was using receiving electrodes placed in furniture, to provide better resolution in living environments.

2.1.5. Data processing

In order to acquire usable data from any digital sensor an analog signal has to be acquired and processed. A simplified typical processing pipeline for this is shown in Figure 2.8. This basic structure is also applicable to the processing of capacitive proximity sensor data. The analog signal is the capacitance of an electric circuit that can be digitized using different methods, e.g. by using the quantized discharge time of the circuit. In the following section some typical steps of raw data processing and high-level processing for capacitive proximity sensors are presented and discussed.

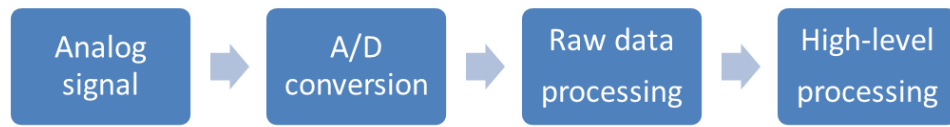


Figure 2.8.: Abstracted sensor data processing pipeline

2.1.5.1. Raw data processing

Raw data processing of capacitive proximity sensor data is primarily intended to compensate for sensor noise and environmental influences. Noise is an inherent property of any measurement system and describes random unwanted data that is added to a signal. Environmental parameters can have strong influence on the signal of a capacitive sensor system. These effecting factors include temperature, humidity, composition of the air, or grounded objects in close proximity. There are numerous additional preprocessing steps that can be taken, such as different multiplexing methods that may be required in some hardware settings, or signal quantization that reduces the outgoing data to a distinct set of values in order to simplify post processing of different applications.

Noise Reduction In order to deal with noise, some sort of filtering is typically applied. Filtering describes a set of methods that attenuate the parts of a signal that are relevant in a given application. In capacitive proximity sensing we are dealing mostly with high-frequency noise that is added to the signal. Therefore, low-pass filtering can be used to deal with this influence. The most typical examples are average filters that take various samples and calculate an average value, and median filters that are sorting a set of samples and select the median element. Each of those filters has a plethora of potential adaptations that are not too specific to discuss in this limited space. Some adaptations are discussed in the descriptions of the specific prototypes.

Table 2.1.: Baseline calibrations terms and methods

Name	Description	Application
Initial calibration	First set-up of baseline at system start, e.g. by taking the average over various samples	Required for any application
Static baseline	Baseline that does not change at run-time	For static environments
Dynamic baseline	Baseline that changes over time	For non-static environments
Drift	Change of system response to environmental factors at run-time	-
Drift compensation	Methods to account for occurring drift, by changing the baseline value	Non-static applications
Recalibration	Change of the baseline value at a specific point in time given a set of rules	Non-static applications

Baseline Calibration A very important aspect of capacitive raw data processing is signal calibration. The generated electric field is subject to changes over time, if either intrinsic parameters change or the environment is

modified. Some specific examples include the electronic components heating up, the environmental temperature changing, or objects being moved in and out of detection range. Additionally, the previously described parasitic capacitance can be compensated to achieve a signal that comprises only relevant information. Therefore it is essential to have a well-calibrated and adaptive baseline; that is the sensor signal generated in the environment without the presence of any object that shall be detected. Again, there are numerous methods to adapt and configure the baseline. I have collected a few common terms and methods and give some pointers regarding their application. The results are shown in Table 2.1. If a dynamic baseline is used, a set of rules will have to

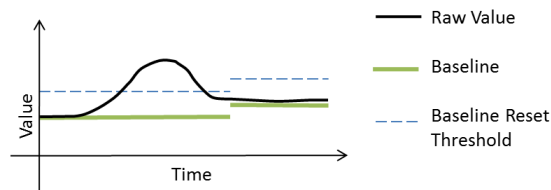


Figure 2.9.: Example of baseline reset using a threshold rule

be defined that determines at which points in time the baseline has to be recalibrated and what specific methods should be used. One simple example is to define a threshold level that triggers a baseline calibration, as shown in Figure 2.9. The raw signal is above the threshold, indicating the presence of a detectable object. Afterwards, it falls back down below the threshold, yet stays for a certain time above the baseline. This could be attributed to changed environmental influences or the presence of noise. In this case a reset of the baseline after a certain amount of time can be triggered.

2.1.5.2. High-level processing

High-level processing assumes that there are calibrated (and preferably normalized) sensor values that are used in any further steps. The goal of any capacitive sensing application is the acquisition of information about a detectable object, e.g. its current position, the material used, or the shape. In order to get this information, we need to use knowledge about the object and intrinsic properties of the sensor system. In this section I will discuss methods to combine data from various sensors using the system properties, how to track the position of an object using different methods and how to recognize specific features. An overview of the methods in abridged form is given in Table 2.2.

Sensor data fusion Sensor data fusion in its most general terms describes "the theory, techniques and tools which are used for combining sensor data, or data derived from sensory data, into a common representational format" [Mit07]. Using the combined information from various capacitive proximity sensors it is possible to generate high-level information that exceeds the capabilities of a single sensor. Two methods can be distinguished. The first is uniform fusion that uses the information from all involved sensors in a common way. The second is heterogeneous fusion that combines groups of involved sensors that serve multiple purposes, yet are attached to a single system. A simple example for the latter would be a single large electrode sensor that detects the presence of a hand from a farther distance and then a combination of various small electrodes that track single fingers.

Sensor data fusion often requires taking into account some additional information that we possess about the system. A classic example is the precision or bias of the sensor. Various methods, e.g. the class of Kalman filters, use weighted information from several sensor sources [WB95]. If we know that a certain sensor is only half as precise as another one working in collaborating, the weighting factors can be adapted accordingly.

Table 2.2.: Overview of high-level processing methods for capacitive proximity sensors

Name	Description
Sensor data fusion	Combining sensor data into a shared representational format
Uniform fusion	Sensor data fusion that combines all data into a single common format
Heterogeneous fusion	Sensor data fusion that combines groups of data to serve multiple purposes
Object tracking	Continuous identification of an object within the systems range
Single object tracking	Methods to realize object tracking for a single detectable object
Multiple object tracking	Methods to realize object tracking for multiple objects
Feature recognition	Identifying certain parameters of an object within the system range

A very important additional information that can be used when fusing data of capacitive proximity sensors, is the geometric layout of the system. This describes position and size of all electrodes that are integrated. Using this information is crucial when trying to localize an object. A simple example would be applying a weighted average algorithm on a set of sensors. In order to determine object location relative to the plane a weighted average algorithm is used. The linear object location \bar{x} is calculated using the sums over sensor positions x_i and sensor values v_i as weight:

$$\bar{x} = \frac{\sum_{i=1}^n v_i x_i}{\sum_{i=1}^n v_i} \quad (2.6)$$

Using similar methods we are able to determine the location of multiple objects or additional dimensions of the position. However, it is possible to use other information in the fusion process as well. The electrode material may result in a different response and thus should be treated differently in a fused data representation and can be weighted. Another example is the shape of the electrode that may result in different responses. How to apply sensor data fusion is strongly depending on the application and the desired common representation that is most suitable for subsequent calculations.

Object tracking In the previous section about sensor data fusion I have shortly discussed a method to determine the linear position of a single object using a linear array of capacitive proximity sensor. This is a basic example of a group of methods associated to object tracking. In computer vision applications they can be defined as "the problem of estimating a trajectory of an object in an image plane as it moves around a scene" [YJS06]. The analogy to capacitive applications is viable, if we consider a 3D scene and a distinct interaction space instead of a scene.

Capacitive proximity sensors allow the detection of conductive objects within their range. However, as this presence is determined indirectly using the influence on an electric field it is not possible to get a direct association

between the actual distance between sensor and object and the resulting sensor value. The created electric field is only analytically descriptive for very specific, theoretic classes of objects [Bax96]. Nonetheless, we are able to get a relative distance measurement. If we combine this proximity value using geometric information about the electrode locations, we can infer the relative position of an object in the sensing area. The weighted average method presented in the previous section is one option for relative positioning. Another method is trilateration, which is similar to many radio-based localization applications. It uses the known location of three or more points and the known distance to the position to be determined. In case of capacitive proximity sensing this position is determined relative to the electrodes as there is no absolute distance measurement. A more complex example for direct calculation was presented by Smith, who formulated the issue of detecting multiple objects as a forward problem and used numerical methods to estimate the position and orientation of two hands [Smi99]. A

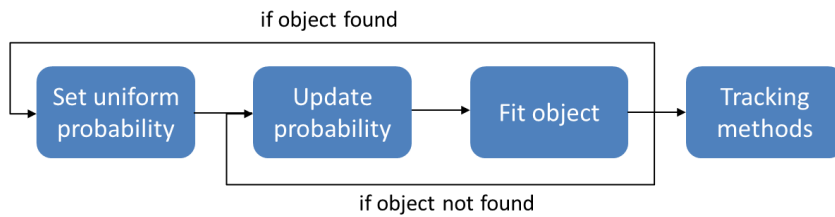


Figure 2.10.: Generic pipeline of probability based methods of capacitive proximity sensing

second class of methods to track objects is not relying on direct geometrical calculations but instead formulates a numerical solution to a probability distribution. The initial assumption is that the probability of an object to be at a certain point in the detection area is uniform. The methods then follow a few basic steps, as shown in Figure 2.10. At first the probability is updated based on the current sensor readings and a priori knowledge about the system. Afterwards, it is attempted to fit the objects into the resulting probabilities. This step may or may not work, meaning that it can result in no object found. In the latter case the process will have to start at the beginning. If an object is found the probability update may use the current object location in the update algorithm, thus starting with a non-uniform probability distribution.

One example for probability-based object recognition using capacitive proximity sensors was presented by Grosse-Puppenthal et al. [GPBKK13]. Using a model suggested by Smith the basic idea is using the assumption that an object may be present anywhere, remove regions where no objects can be present and then fit an object into the remaining space. This method additionally uses particle filters to track object locations over time. This also allows tracking multiple objects. Another method, introduced by Le Goc et al., uses random decision forests to determine the location of a hand [LTI*14].

Throughout the years various methods have been suggested for supporting multi-object tracking using capacitive sensors. Touch screens often use inversion of the sender signal to reliably detect the positions of multiple points. However, this method can't be used without adaptations in proximity applications [WF07]. Some of the previously presented methods support the tracking of two or more objects. There are still various limitations, particularly if not only the object location, but also various other features such as rotation should be tracked. This is still an area of ongoing research, leading to the next area of high-level processing - feature recognition.

Feature recognition Feature recognition is primarily used as a term in image processing, traditionally in computer-aided design applications. It allows to recognize specific geometric properties of an object but also picture analysis, e.g. in facial recognition [HPR00, BHK97]. In the domain of capacitive proximity sensing, feature recognition can be defined as the acquisition of non-location information from any detectable object. An

Table 2.3.: Feature recognition methods

Name	Description
Data-driven methods	Directly associate input data to output features using various methods, e.g. machine learning and training data
Model-driven methods	Input data is manipulating a pre-defined model of the system that is latter mapped to the output
Neural networks	Computational models using a network of neuron-like objects that are often used in machine learning
Pattern recognition	Methods that look for certain patterns in a set of input data

important feature in industrial applications is the material of an object [Bax96]. With regards to recognizing additional features a system was presented by Wimmer et al. - Thracker [WHKS06], a prototype augmenting a regular monitor with capacitive proximity sensors. In addition to recognizing hand position the system is able to detect grasp gestures, which can be used to select items on the screen and perform pick and drop operations. Capacitive sensors can also be used to distinguish between persons and a children's seat on the passenger side of a car [GZBB09]. The methods to recognize the features can be divers, ranging from typical machine learning algorithms, to model-based approaches. An incomplete list is given in Table 2.3. In order to keep this work contained I will refrain from a deeper discussion at this point.

2.2. Capacitive proximity sensing applications

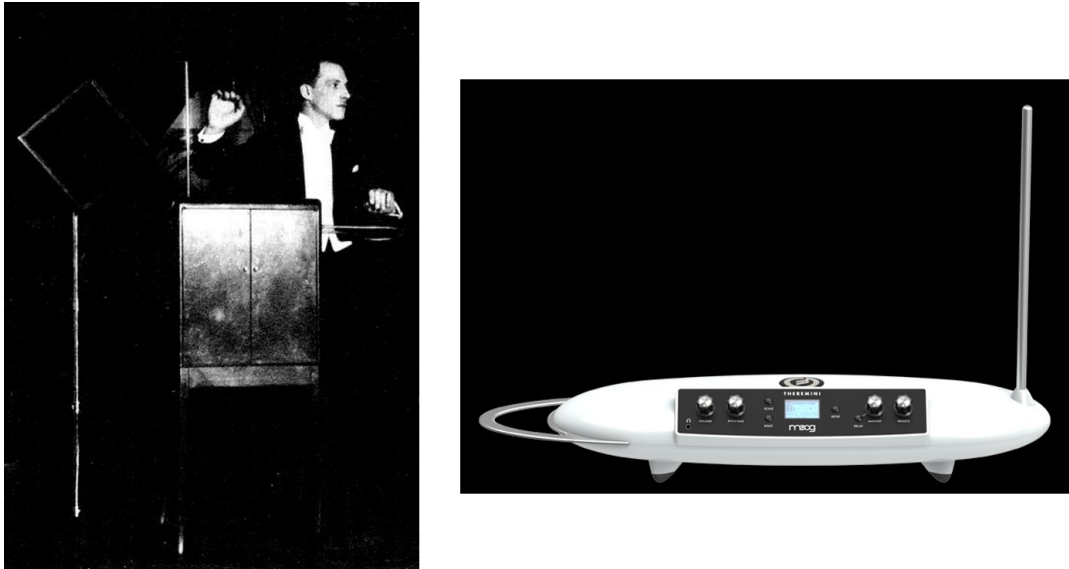


Figure 2.11.: *Left*: Leon Theremin playing his eponymous electronic musical instrument [Gli00]. *Right*: The Theremini by Moog Music Inc., released in 2014 [Moo14].

In the last decades of the 19th and the beginning of the 20th century a considerable number of inventors and scientists performed research on the application of electric systems, sparking innovations such as electric lighting, electric motors, telegraphy, and radio communication. Lev Sergeyevich Termen or Léon Theremin in the American naming was a Russian inventor most famous for designing the eponymous theremin. This early electronic musical instrument could be played without touch. One hand is controlling the pitch and the other the volume by changing the distance to an antenna. Initially designed as a motion detector, this device is transferring the influence of the human body on an oscillating electric field to an audible sound [Gli00]. Léon Theremin can be seen playing the instrument in Figure 2.11 on the left. This instrument is still in production to this day, with the electronic music instrument company Moog releasing a new variety that simplifies the sound production by fitting the distance to a sound on a specific musical scale [Moo14]. The instrument is shown in Figure 2.11 on the right.

Electric field imaging was a research focus at the MIT in the 1990s. A research group in the Media Lab division including Joseph A. Paradiso, Thomas G. Zimmerman, and Joshua Smith designed various sensing devices and evaluated various applications in the domains of human computer interaction, smart appliances and reactive systems. They drew inspiration from biological precedents - various species of fish, such as Gymnotoidei can sense their surroundings using electric fields [Smi99]. The changing currents created by objects with a different dielectric constant from water can be registered and thus used to avoid obstacles, even if no light source is available. Accordingly, the group named some of their prototypes after this biological precedent, including the LazyFish and School of Fish [Smi99]. The research group created various different applications in the domains of human computer interaction, smart appliances and reactive systems.

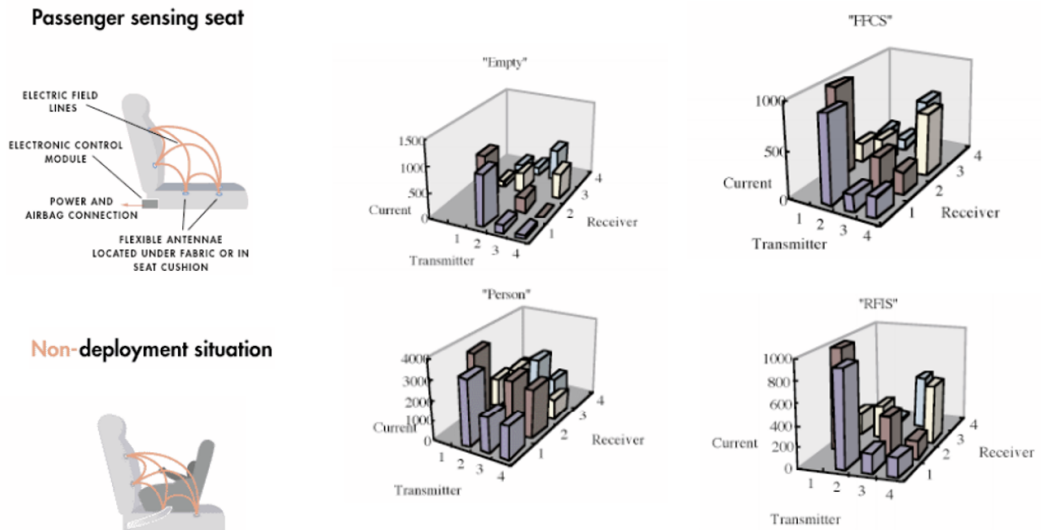


Figure 2.12.: *Left*: Concept view of passenger seat set to deploy or not deploy airbag. *Center*: Sensor readings for empty seat and adult person. *Right*: Sensor readings for front-facing child seat (FFCS) and rear-facing child seat (RFCS). [Smi99]

In collaboration with NEC, a smart passenger seat was created that incorporates capacitive sensors operating in shunt mode to detect if an infant seat is currently present on the passenger seat of car [Smi99]. The underlying challenge is that airbag deployment should be prevented in this case to prevent potential injuries to the infant. The seat is able to distinguish four different states, “No passenger”, “adult passenger”, “front-facing infant seat” and “rear-facing infant seat”. It uses four sending and four receiving electrodes and classifies the situation according to the current readings - the concept and readings of the sensors in the different situations are shown in Figure 2.12.

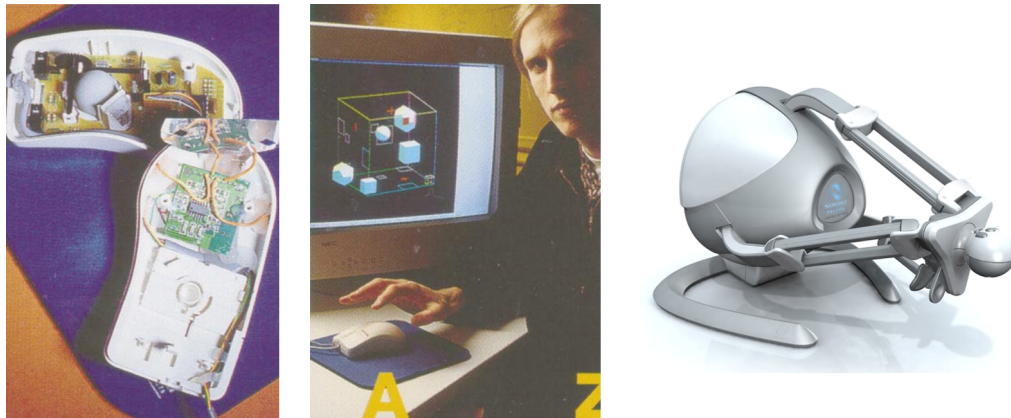


Figure 2.13.: *Left*: LaZmouse innards *Center*: Joshua R. Smith using LaZmouse [Smi99]. *Right*: Novint Falcon 3D input device [Nov14]

Another prototype is the LaZmouse that extends a regular mouse with shunt mode capacitive sensors, having one transmitting and two receiving electrodes. It measures the proximity between the heel of the hand from the mouse surface, thus allowing the fingers to remain in the common position and the mouse to be moved around [Smi99]. Effectively this creates an input device with three degrees-of-freedom, enabling to perform interactions with a mouse that would usually require a more specialized 3D input device, such as the Novint Falcon that tracks the movement of the moved interaction sphere in three dimensions [Nov14]. Figure 2.13 shows the electronics inside the LaZmouse on the left, the inventor using the device and a graphical representation in the middle, and on the right the Novint Falcon.

In another work Paradiso et al. presented numerous applications for capacitive proximity sensors, including smart furniture devices [ZSP*95]. They propose a smart table, comprised of a single transmitter electrode and two receivers that is able to track the position of a hand in two dimensions, a chosen dimension on the table and height. It may be used as gesture input device or to augment video desk applications. They also installed the system in a room, whereas the floor is a single electrode and there are four receivers located on the walls. This allows to infer the location of a person, based on relative signal strength. An additional system in this work is the smart chair that is using a single transmitter in the seat and four receivers in the armrests and headrest. It allows to navigate through various audio channels based on head and arm movements [SM95].

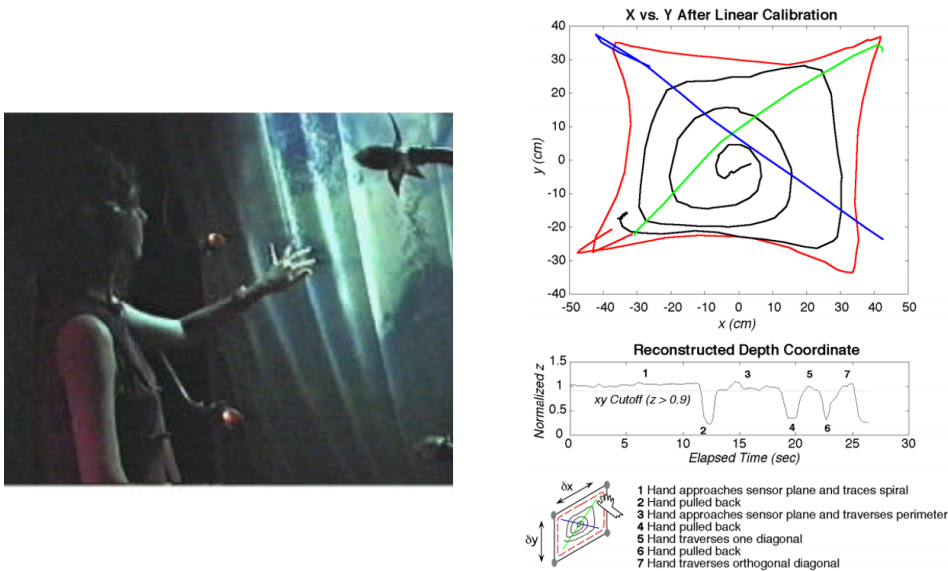


Figure 2.14.: *Left*: Person interacting with the gesture wall *Right*: Air drawing results, depth estimation results and associated movements on bottom [SWD*98].

A final prototype of this group I will present is the Gesture Wall. This is a large interactive multimedia wall, designed for public appearances [SWD*98]. A plate on the floor in front of the screen is acting as transmitter and four receiving electrodes are placed in the system that protrude from the edges of the projection area, as shown in Figure 2.14. It supports interactive experiences, such as drawing in the air, controlling different audio streams and an interactive video clip.

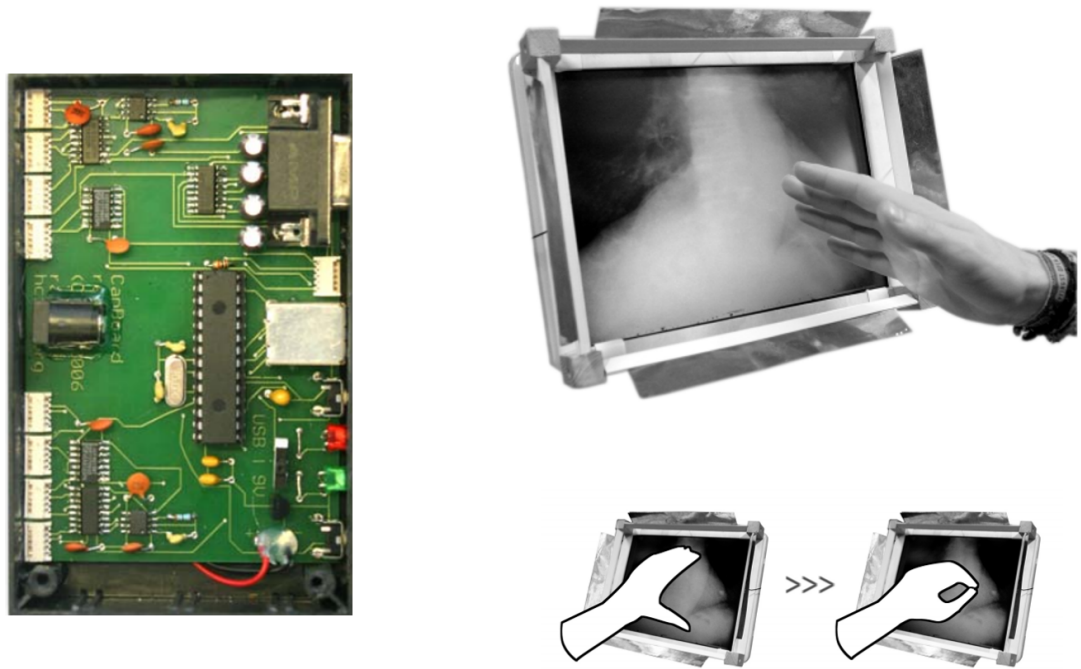


Figure 2.15.: *Left:* Prototype of CapToolKit [WKBS07]. *Right:* Thracker prototype and visualized grasping gestures [WHKS06].

Another group that was active in capacitive proximity sensing was located at the University of Munich. Raphael Wimmer and colleagues revisited capacitive sensors in the scope of human-computer interaction. They created CapToolKit, a capacitive sensing rapid prototyping toolkit that allows interfacing eight capacitive proximity sensors and enables a quick design and testing of new applications [WKBS07]. They also created Thracker - a display augmented with four capacitive proximity sensors that allows to detect the position of the hand in front of the screen and supports performed pick-and-drop gestures [WHKS06]. Both devices are shown in Figure 2.15.

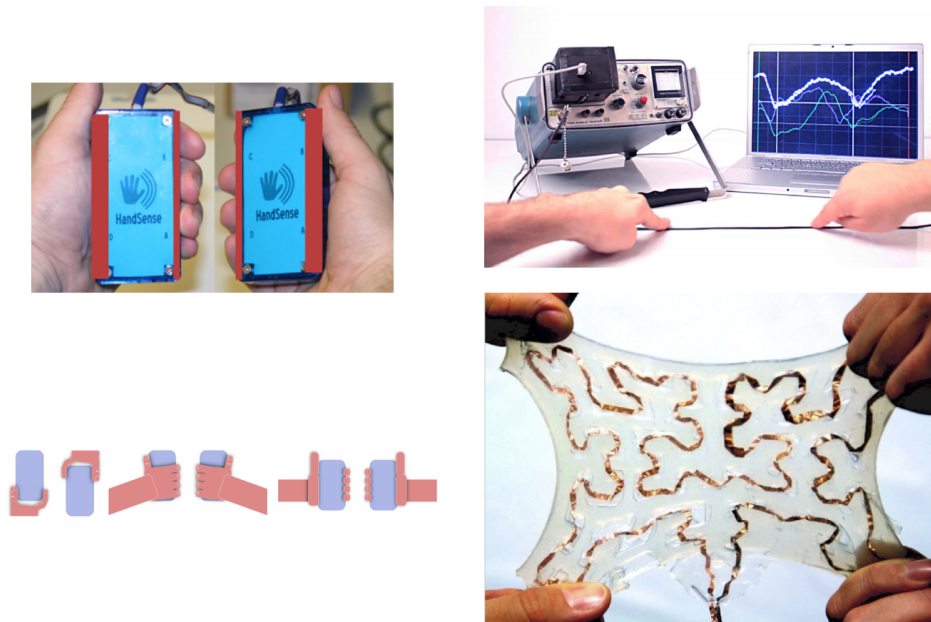


Figure 2.16.: *Left*: Prototype of HandSense and supported grasping types [WB09]. *Right*: Setup of time domain reflectometry sensing and example of stretchable material [WB11].

In other works they also presented capacitive sensing to allow discriminating the ways an interaction device is held, including distinguishing left and right hand or proximity to a body part [WB09]. This enables graphical interfaces to be adapted, based on user-handedness, grasping style and spatial cues that can be acquired from this device in collaboration with other sensors. Supported grasping styles and a prototype are shown in Figure 2.16, on the left.

A final work of this group was concerned with exploring the potential of time domain reflectometry for human-computer interaction [WB11]. This technique is sending a short electrical pulse into an electric conductor and measures the time until the signal returns. Originally intended for discovering defects in long cables, such as transatlantic phone lines, high-sampling rates enable to also detect the presence of grounding objects close to much shorter conductors. Wimmer and Baudisch use an image analysis on the screen of an older reflectometer to enable applications, including location tracking, touch detection and stretchable materials. The setup and an example of stretchable materials are shown in Figure 2.16, on the right.

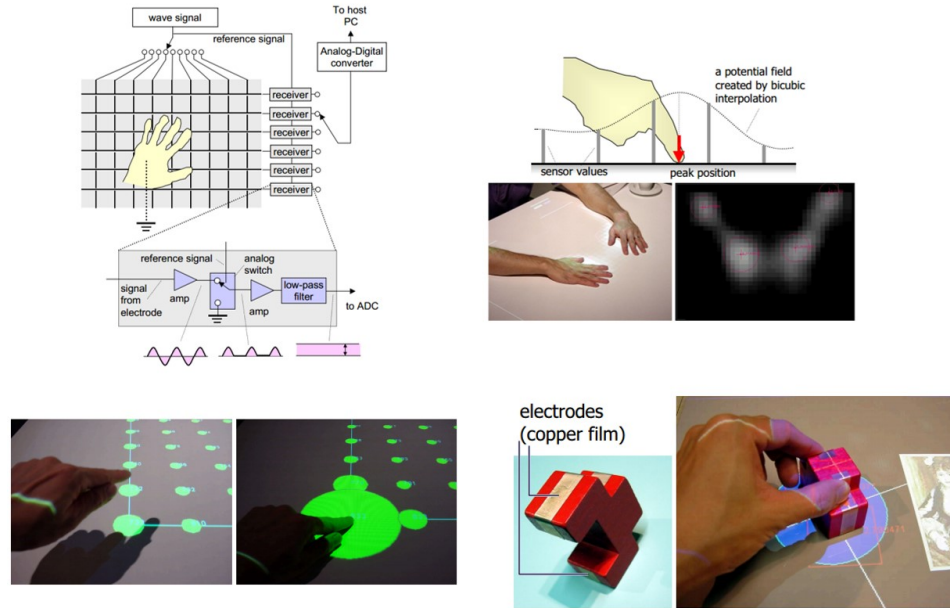


Figure 2.17.: *Top left:* Technical configuration of SmartSkin. *Top Right:* Bicubic interpolation method to detect the peak of the potential field created by hand proximity. *Bottom left:* Visualized sensor effect of hand hovering in 10cm distance and touching the surface. *Bottom right:* Capacitance tag having a specific conductive pattern that can be identified [Rek02].

Jun Rekimoto presented SmartSkin, an interactive table based on capacitive proximity sensing based on a number of sensors configured in shunt mode [Rek02]. They are able to detect hands and fingers at a distance of up to 10cm, the sensor response being shown in Figure 2.17 on the bottom left. He investigated two different prototypes based on this technology and developed an image-based method to detect the position of hand or fingers above the surface, as shown in Figure 2.17 on the top right. Additionally, capacitive tags were introduced that enable tangible interaction by detecting the type of object the user is currently touching on top of the surface. It analyzes their specific capacitance pattern, resulting from a difference in shape and layout of conductive material covering the object, shown in Figure 2.17 on the bottom right.



Figure 2.18.: *Top*: Capacitive profiles of different gestures on a door knob [SPH12]. *Bottom Left*: User identification using capacitive fingerprinting [HSP12]. *Bottom Right*: Botanicus Interacticus prototype and interaction concept [PSLS12].

Another research group that has been recently active researching capacitive sensing systems is centered around Ivan Poupyrev at Disney research. They presented Touché, proposing a novel sensing technology called swept-frequency capacitive sensing [SPH12]. Instead of relying on a single excitation frequency, like typical capacitive sensors, this measurement system is sweeping a larger range of frequencies. This allows to get more data points from a single electrode. Many systems react differently to changing frequencies, including the human body and its complex structure of bone, muscle, fat and other tissues. The result is a capacitive profile comprised of values at different frequencies, as shown in Figure 2.18 on the top. Using these profiles it is possible to distinguish numerous events in different applications, e.g. the different ways in which a mobile device can be held. Other applications proposed in this work were sensing of body postures on a table, on-body gestures using wrist-worn sensors, interaction with a body of water and grasping a door knob using different gestures. Building on this work Harrison et al. proposed a system using this technology to identify the users of a touch screen with a capacitive fingerprint [HSP12]. This allows to distinguish the touches of different users, thus enabling a variety of applications on collaboratively used touch screen devices. Examples are gaming and multiuser drawing tools, as shown in Figure 2.18 on the bottom left. An interesting suggestion are personalized undo stacks that allow a more seamless collaboration. A final system is the Botanicus Interacticus that uses the same technology to realize interactive installations using plants [PSLS12]. It is possible to distinguish the location a certain plant is touched and associate it to different input events. In this first system it was used in an interactive art installation that is shown in Figure 2.18 on the bottom right.

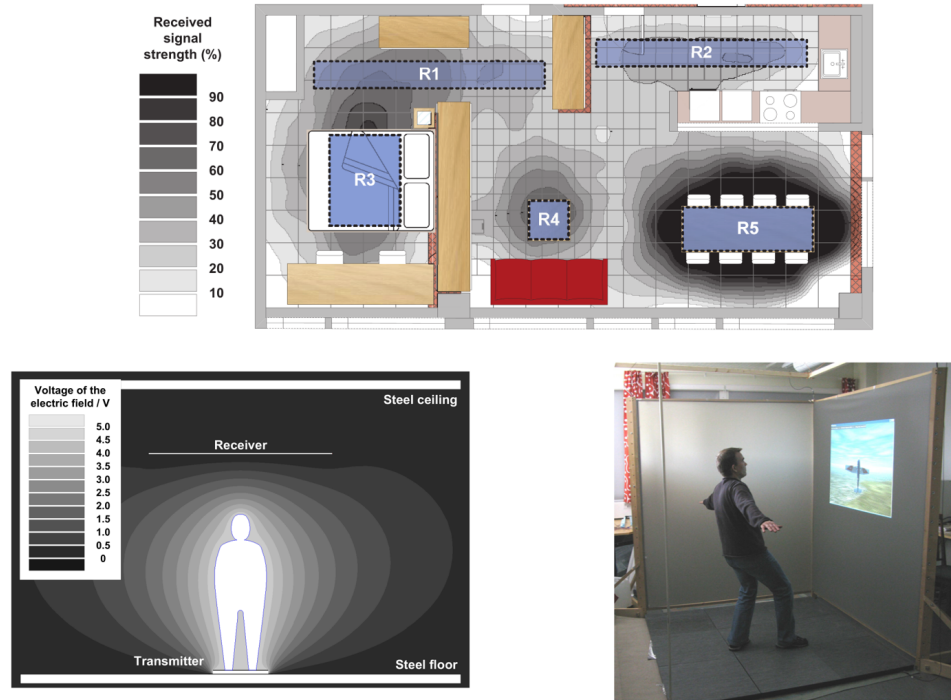


Figure 2.19.: *Top*: Measurement of TileTrack resolution with five integrated receivers [VV12]. *Bottom Left*: Simulation of transmitted field by a single person [VKMV11]. *Bottom Right*: Using TileTrack system for interaction [VRV10].

A final research group working with capacitive proximity sensing that I would like to present is located at the Tampere University of Technology. Valtonen et al. have been working on a capacitive proximity sensing systems tracking the position of users in a room [VMV09, VV12]. A transmitting electrode is placed under the floor and several receiver electrodes are hidden in the walls or other objects of the room. The system therefore is using the presented transmit mode, coupling the users present in the environment to an electrode in the floor and measuring the effect on the receiver electrodes. A downside of this system is the required proximity to a wall electrode, making application in larger rooms impossible. Accordingly, an addition was presented that hides the receiving electrodes in different items of furniture to provide a more reliable localization in living environments [VV12]. The placement of the receiver electrodes R and the received signal strength in the environment are shown in Figure 2.19 on the top. An extension of this principle is using a similar electrode configuration to determine height and posture of a person between a transmitter electrode on the bottom and a receiver electrode on the ceiling [VKMV11]. A simulation of the electric field voltages created in this setup is shown in Figure 2.19 on the bottom left. The taller a person, the higher the resulting signal. Accordingly, a sitting or lying person will have a much smaller effect and can be classified. A similar setup was also used to identify different gestures performed in an interaction area, as shown in Figure 2.19 on the bottom right [VRV10]. Using a single transmitter and four wire receivers in the corners it is possible to track different body postures and control some example applications.

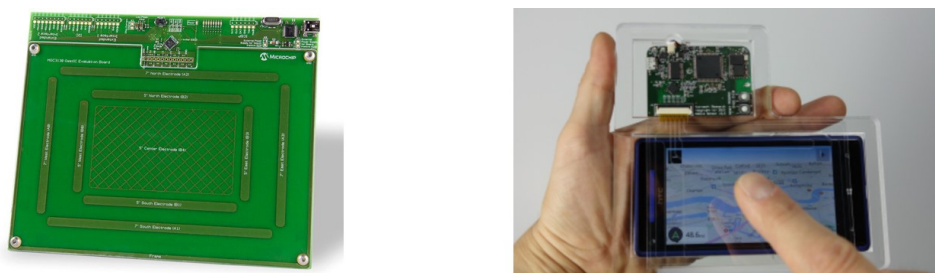


Figure 2.20.: *Left*: GestIC sabrewing prototyping board [Mic13]. *Right*: Transparent interaction device based on GestIC [LTI*14].

Recently capacitive gesture interaction has also captured the interest of industry. GestIC by Microchip Technologies Inc. is a controller for electric near field 3D tracking based on capacitive proximity sensing [Mic13]. A prototyping board is shown in Figure 2.20 on the left. Using a set of several electrodes and on-board processing it is capable of tracking the 3D position of a hand and provides gesture recognition at a distance between 0cm and 15cm. It is based on shunt mode sensing and has a number of different available electrode configurations. This system was used in a recent publication by Le Goc et al. [LTI*14]. They used a transparent ITO electrode configuration mimicking the prototyping board configuration and presented a novel machine learning method that improves the precision of the object localization. They presented a prototype system, augmenting a mobile device with 3D gesture interaction, as shown in Figure 2.20 on the right.

2.3. Sensor systems in smart environments

In the most general definition a sensor is a device that transforms a physical property into an observable signal. This definition includes traditional systems such as mercury-based thermometers or hair-based hygrometers. Yet nowadays, we are usually considering digital sensors that transfer the measured property to a binary signal that can be further processed by computing devices.

A common variety is the smart sensor that provides additional functionality beyond generating a correct sensing signal [Fra13]. The main goal is to simplify installation and maintenance of distributed sensing systems by having processing close to the measurement device. Early considerations in this domain were put to the standard family IEEE 1451 - IEEE Standard for a Smart Transducer Interface for Sensors and Actuators between 1997 and 2007 [iee07]. An additional concept is the Virtual Sensor that includes digital signal processing and conditioning and therefor abstracts the processing steps from devices interfacing the sensor. The number of available sensors is very high, but it is possible to restrict them based on their application domain. Lewis, as well as Cook and Das [Lew04, CD07] have proposed a collection for smart environments focused on wireless sensor networks. The overview is shown in Table 2.4.

This sensor categorization is based on the property to be measured and is independent from the specific measurement technology. Physical properties, such as pressure, temperature, humidity and flow, can also be noted as environmental properties. They are measurements that determine the state of the smart environment, e.g. temperature in different rooms, or the current water usage. Motion properties denote the movement parameters of actors in this environment and can refer to both humans and machines. Angular velocity is important in self-localization of robots in an environment. Contact properties groups the different types of interaction between surfaces in the smart environment and actors. Presence as a group is similar to motion parameters, but does not require a series of measurements for tracking an actor. Biochemical sensors enable measuring the presence of specific chemical compounds in the environment and are most suited for measuring pollution or air quality. Finally, identification

Table 2.4.: Sensors for smart environments [CD07]

Properties	Measurand
Physical properties	Pressure, temperature, humidity, flow
Motion properties	Position, velocity, angular velocity, acceleration
Contact properties	Strain, force, torque, slip, vibration
Presence	Tactile/contact, proximity, distance/range, motion
Biochemical	Biochemical agents
Identification	Personal features, RFID or personal ID

of actors allows to provide personalized services and can be realized with different methods ranging from tags to biometric systems.

While this listing provides a decent overview of sensing properties in smart environments it is abstracted from sensor technologies. Various types of sensors, including capacitive proximity sensors, allow us to detect multiple of these properties and thus providing a higher flexibility. Therefore, it is possible to provide an inverse listing of sensor technologies that allow measuring different properties. A short overview of sensor technologies with this capabilities and that are commonly used in smart environments is given in Table 2.5.

In the following sections I want to give an overview on how these sensor systems are used in this domain, in order to provide a basis for the benchmarking model that will be introduced in Chapter 3.

Table 2.5.: Sensing technologies and measured properties

Technology	Properties
RGB cameras	Motion, Presence, Identification
Infrared cameras	Motion, Presence, Contact
Ultrasound sensing	Motion, Presence, Contact, Identification
Microphone arrays	Motion, Presence, Contact, Identification
Radiofrequency sensing	Motion, Presence, Identification

2.3.1. RGB cameras

A RGB camera is an image processing device that processes light in the visible spectrum, similar to the human eye. Modeled after the retina it has three distinct color channels - red, green and blue. There are different methods available to distinguish these channels from visible light, such as Bayer filters (Figure 2.21) in front of a single sensor or using multiple sensors behind a prism. The usage of cameras in smart environments is very common. I will present five different examples and afterwards will specify how they are linked to the properties that were defined previously.

Tabar et al. have been using a combined system of cameras, RF transmitters and wearable sensors in a home care scenario [TKA06]. The cameras are used to improve the accuracy of the accelerometer-based fall detection

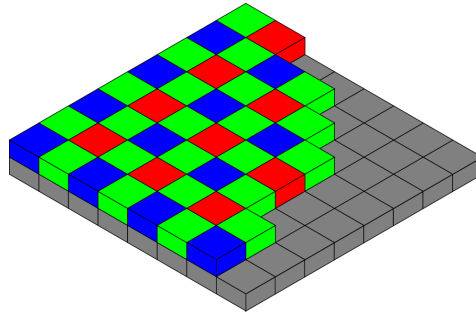


Figure 2.21.: A bayer pattern on a sensor in isometric perspective [Wik06]

by eliminating false positives. Once a fall event occurs an algorithm tracks the posture of detected humans in the scene (Figure 2.22 - left). They used an edge detector to distinguish the human body from other objects and applied a heuristic to differentiate lying and standing. Additionally a face detector was used to improve the recognition of human objects. Combining this with information from the fall detecting sensor and a RF based localization system they were able to achieve a good reliability in eliminating false positive alerts.

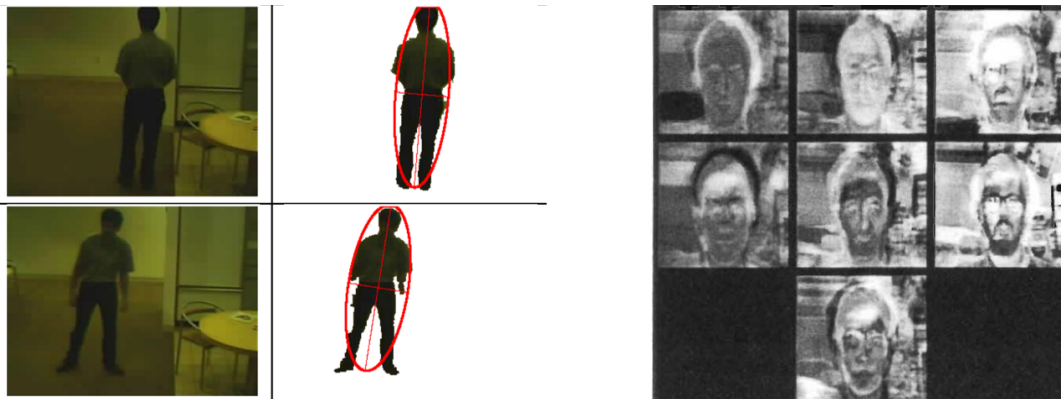


Figure 2.22.: *Left*: Tracking of body masks using cameras [TKA06]. *Right*: Eigenfaces created from input picture set [PC00].

Pentland and Choudhury provided an overview of vision-based face recognition systems in the domain of smart environments [PC00]. The systems are able to identify users and recognize facial expressions. The proposed applications in smart environments include personalized shopping experiences based on customer recognition, behavior monitoring in child care facilities and emotion-aware systems that react to the user's current awareness. The described techniques include PCA-supported, eigenvector-based classification, face-based localization and systems based on local feature analysis (Figure 2.22 - right). Newer systems are able to operate well in unconstrained environments that include varying expression and illumination, ageing of persons, occlusion and disguise [WYG*09].

An example for a novel image processing method that is useful in smart environments was presented by Vu et al. in 2012 [WRS*12]. They are using temporal variances of pixel values to exaggerate spatial movements and color changes that would typically be invisible to the naked eye. The method is called Eulerian Video

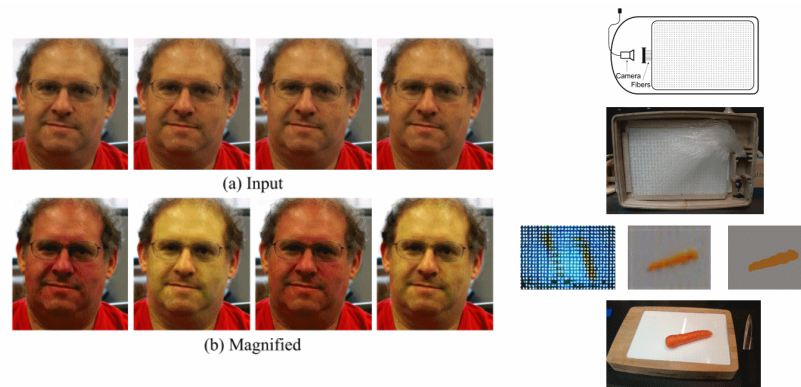


Figure 2.23.: *Left*: Eulerian Video Magnification to attenuate the human pulse with original (a) and amplified (b) video sequence [WRS*12]. *Right*: FoodBoard schematics (top), underside view (second row), original, reconstructed and segmented image (third row) and final system (bottom) [PJS*13].

Magnification and uses a combination of spatial decomposition and temporal filtering applied to adjacent frames. It can be tuned to different time-frequency bands to attenuate different classes of signals. Some of the proposed applications include the tracking of breathing rates of infants by attenuating chest movement, or the tracking of subtle movements, such as vibration in appliances. The example shown in Figure 2.23 on the left is using a magnification of colors, in order to identify the heart rate of a person. The latter can be used for personal health applications, e.g. by integrating the system into the bath room mirror to provide an unobtrusive daily measurement and give the user feedback over a longer period of time.

A final example in this section is the FoodBoard, a smart chopping board that uses image processing to recognize the food items that are put on it [PJS*13]. It is shown in Figure 2.23 on the right. To enable a thin footprint, ambient light is transferred to a camera using glass fibers. The picture is reconstructed and segmented, allowing to identify different items of food that are placed on it. The classification is based on a combination of Fast-Hessian and color histogram feature extractors. Pham et al. were able to distinguish 12 different ingredients with an accuracy between 59% and 93%. The system can be used to support dietary monitoring, give recipe guidance or support visually impaired users in identifying and tracking food.

2.3.2. Infrared cameras

Infrared imaging is based on the same sensors that are used for visible light imaging. The difference is that they are tuned to collect electromagnetic waves of a lower wavelength, which is just outside of the human visible spectrum. This allows for distinct applications, such as thermal imaging, as it is possible to detect heat radiation.

One of the earliest prototypes in Ubiquitous Computing designed by PARC was using the ORL Active Badge, an infrared emitter developed by Olivetti Research Laboratories that was used to identify persons operating in the environment [Wei91] (Figure 2.24 Top Left). In smart environments the most common application is using infrared cameras in combination with infrared light sources. This allows to illuminate spaces without visible artifacts to the user, thus providing imaging capabilities in dark rooms, or very specific conditions that may be required by a certain application. Another very interesting option is to use a specific projection of patterns into the scene. Analyzing the returning infrared light it is possible to infer the depth of specific pixels of the camera (Figure 2.24 Top Right). This variety is called a depth camera. Particularly in the last few years the research



Figure 2.24.: *Top Left*: ORL Active Badge [Wei91]. *Top Right*: Kinect infrared projection [Zha12]. *Bottom Left*: Kinect Fusion reconstruction [IKH11]. *Bottom Right*: Kinect kitchen interaction [Pan12].

in this domain has expanded strongly, sparked by the availability of an affordable depth camera/RGB camera combination - the Kinect by Microsoft [Zha12]. On the following pages I will present various examples of how this device can be used in smart environments to enable different applications in interaction and activity tracking.

Sung et al. have presented a system that is tracking activities of daily life based on the movements of a skeleton that is provided by the Kinect API [SPSS11]. This skeleton model is based on a pose reconstruction algorithm developed by Shotton et al. [SFC*13] and is used in many different Kinect-based applications. The algorithm is using a method called hierarchical maximum entropy Markov model (MEMM). Each activity is considered to be composed of sub-activities. Based on this assumption a two-level graph is generated using dynamic programming. The system was tested with twelve activities performed in an office, a kitchen, a living room, a bathroom and a bedroom. If the person was part of the training set the precision was 84.3% and 64.2% for unknown persons.

A novel method to use the Kinect for fast 3D reconstruction of scenes was presented by Izadi et al. in 2011 [IKH11] (Figure 2.24 Bottom Left). The basic premise is to use a fast registration method to combine point clouds that are generated by the system, in order to continuously extend and optimize the current model of the scene. They are using a GPU-based ICP implementation to track the position of the camera in six degrees-of-freedom. This allows to reliably integrate the different point clouds into a single volume element grid that is used to represent and render the scene. They proposed a number of applications ranging from physical simulation of particles in the scene, to system control based on segmenting and tracking the user's hands and their interaction with arbitrary surfaces in the environment. Figure shows some results of the touch recognition in a reconstructed scene.

Panger has presented a method to use a Kinect in a kitchen to provide touch free interaction [Pan12]. Two different interaction schemes are proposed. The first allows to control the applications with messy hands, the second enables to use other limbs if the hands are currently occupied (Figure 2.24 - bottom right). Three different test applications have been implemented. A recipe navigator that allows to open and navigate through different recipes. A timer that enables setting different alarm times, similar to a kitchen clock and a music player that can be controlled to choose different stations according to preference. A real life test with five subjects also tested installation complexity, which was deemed favorable. There were some concerns in clearly distinguishing commands from random movements performed in the kitchen.

The final system based on infrared cameras is an immersive telepresence system developed by Beck et al. [BKKF13]. Telepresence enables persons to be present as a representation in a remote place. A typical example is video conferences, but advanced system may include robotics or virtual elements. The presented system is using a single Kinect for each participant. A 3D representation is created based on the depth information of the infrared camera and on-the-fly texturing using the included RGB camera. The virtual user representations are put into a shared virtual environment and can interact with each other. Some variations were tested where local and remote users were decoupled, side-by-side or face-to-face. Different tracing and pointing gestures are supported. The supported applications included the exploration of a virtual city.

2.3.3. Ultrasound sensors

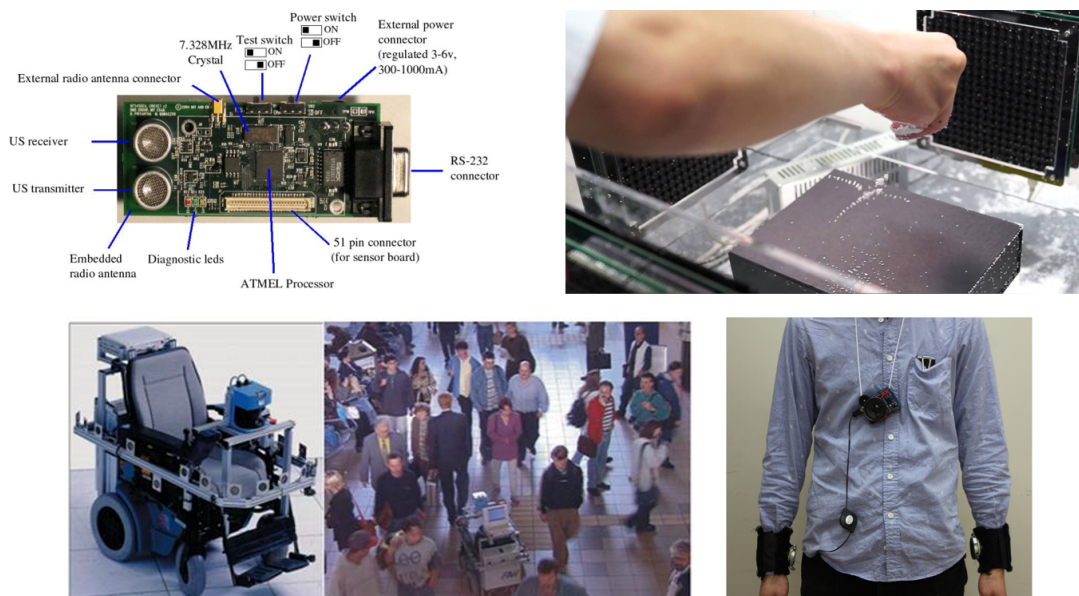


Figure 2.25.: *Top Left*: Cricket Indoor Localization hardware [PCB00]. *Top Right*: Mid-air particle manipulation [OHR14]. *Bottom Left*: Robotic wheelchair MAid with ultrasound range sensors [PSF01]. *Bottom Right*: Activity and context recognition using ultrasound sensors [WTT13].

Ultrasound sensors allow detecting sound wave signals that have a frequency beyond 20kHz and are therefore not audible to humans. Their propagation and reflection properties are similar to audible sound waves, thus the generated measurements can be similar. While there are natural sources of ultrasound waves, the applications in

smart environments rely on active systems that combine sound generators and sensors that measure the resulting signal. By measuring the time distance between sending the signal and receiving a response, it is possible to measure distances between the sender and different objects. If various receivers are used it is possible to localize the sound source, making ultrasound sensing a popular candidate for indoor localization systems. I will present four different examples on how ultrasound sensors are used in smart environment applications.

The Cricket developed by Pryantha et al. is an example for an indoor localization system based on a badge attached to the tracked object [PCB00]. The badge is periodically sending signals to a set of beacons that determine the distance and calculate a location. Initially, it was supposed to solely rely on radiofrequency signals, but was modified to use a combination of RF and ultrasound. The rationale of this decision is the considerably slower speed of sound that simplifies measuring the time-of-flight and in consequence allows for more precise distance measurements. This also leads to a higher precision of the localization algorithm. Additionally, the system uses some novel methods to deal with interference and multipath issues that are caused by superposition of reflected signals. As potential applications they propose space-dependent services that are provided as the user is identified in a certain region, and guidance scenarios on a floor plan. The hardware is shown in Figure 2.25 - top left.

The robotic wheelchair MAid (Mobility Aid for Elderly and Disabled People) was designed to support and transport people with limited motion skills [PSF01]. It is based on a commercial wheelchair that has been equipped with an intelligent control and navigation system. The system includes an ultrasound-based range finder that allows to detect obstacles in the path and circumnavigate around (Figure 2.25 - bottom left).

Watanabe et al. investigate the role of ultrasound sensors in recognizing activities and gestures of a user [WTT13]. The system is comprised of a microphone attached to a necklace, and two speakers that are attached to each wrist. Based on the acquired volume and evaluation of the Doppler Effect, it is possible to determine both distance of the wrists from the neck and the speed of movement (Figure 2.25 - bottom right). Watanabe et al. want to determine if this system allows for similar performance compared to other body-worn systems based on accelerometer and gyroscope data. Additionally it was evaluated if external microphones can perform as well as the neck worn microphone. The system was able to recognize 87% of activities and gestures in a set of 10 test persons if no influencing sound was present. The ultrasound did also improve results if environmental noise was present.

A recent project at the University of Tokyo evaluates the potential of ultrasound in manipulating small particles in free-air [OHR14]. Using standing waves to create sound pressure nodes it is possible to apply a force to small particles that is sufficient to counteract gravity. Using a set of ultrasonic phased arrays it is possible to create these sound pressure nodes at arbitrary positions in three dimensions (Figure 2.25 - top right). Ochiai et al. use this to move small objects around and investigate required object properties and their floating properties. So far the moved items have to be very light and smaller than 2 mm in diameter, resulting in the usage of polystyrene. The technology is also able to hold and move small amounts of fluid. Suggested applications include object manipulation in microgravity environments and projected haptic feedback systems.

2.3.4. Microphone arrays

Microphones are signal receivers that are tuned to detect sound frequency ranges that are audible by humans. Typically, they consist of a piezo element that transfers vibration to an electric current, which is subsequently amplified. Most human activities produce some kind of sound. As opposed to the presented ultrasound system, microphones typically operate passive. There is usually no dedicated signal source, but instead naturally occurring sounds are recorded. By combining microphones, or arrays thereof, with data processing systems that are aimed at analyzing specific sound patterns, it is possible to get information about human activities. Looking at

2. Related Work

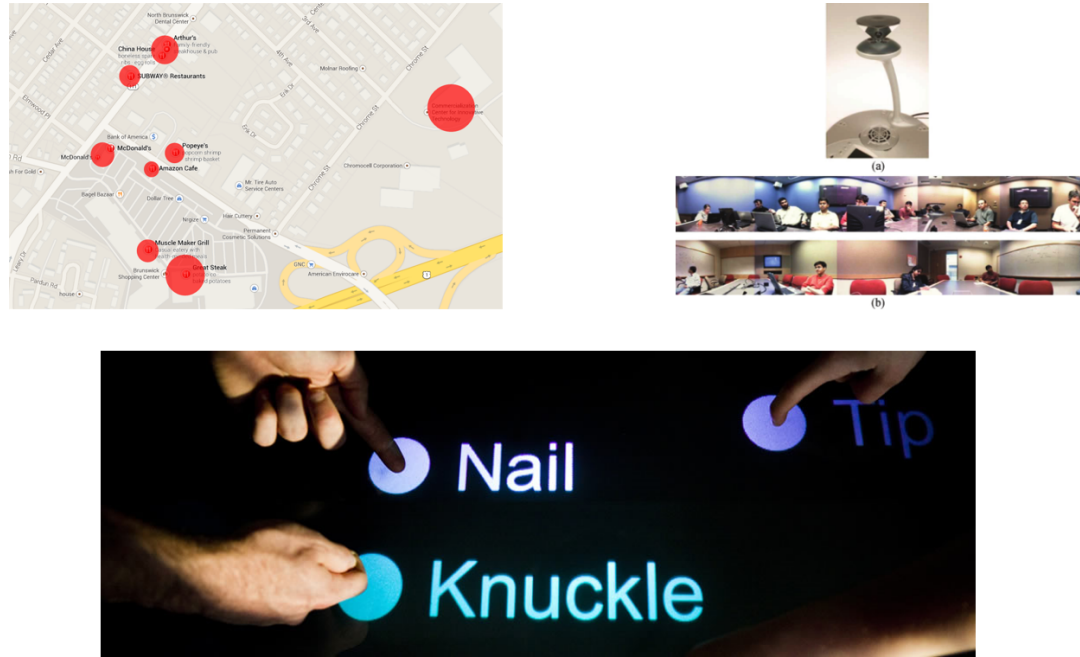


Figure 2.26.: *Top Left*: Visualization of speaker count in different areas [XLL*13]. *Top Right*: Directional microphone and conference room for speech source localization [ZFBZ08]. *Bottom*: TapSense detecting different tap events [HSH11].

smart environments there are numerous use cases that can benefit from microphones as sensors. In this section I will present four different systems that cover a large variety of different applications, ranging from breathing rate detection to estimating the number of speakers in large groups.

Detecting the breathing pattern of a person has several applications in smart environments. Apart from medical purposes that require detecting abnormal breathing in risk groups, it is also possible to track training progress or provide a measure for the current attention level in affective computing. Corbishley et al. investigated very small microphones in mobile devices that detect the breathing rate [CRV08]. The algorithm is designed to be applicable on single ICs, allowing for miniaturization and energy efficiency. Even with the presence of noise the combined score for true negatives and true positives was as high as 91.3%. Using small and energy efficient systems also enables unobtrusive applications in non-mobile environments, e.g. placing such a system close to the bed to detect the breathing rate while the user is sleeping.

Collaborative applications are an important aspect of smart environments, e.g. to link together meeting places at different places, similar to the telepresence application presented earlier. If a multitude of speakers is present it gets increasingly difficult to provide a system that enables proper speech transmission for all participants. Using an array of microphones it is possible to focus the attention on the person currently speaking and filter out environmental noise. A project at Microsoft Research investigated using a maximum likelihood of two known techniques, beamforming and speech source localization, to enable a reliable speaker selection [ZFBZ08] (Figure 2.26 - top right). Additionally, the framework enables a suitable adaptation even if directional microphones are used that are placed close to each other. The method provides a real life accuracy of more than 90%.

A fairly new work was performed by Xu et al. [XLL*13], called Crowd++. Their idea is to use smartphone microphones to identify the number of speakers in crowded environments. Such system could be used to estimate the number of persons in a given place and potentially react quickly if a crowd panic may occur. The method is based on an unsupervised machine learning classification of short audio parts that are picked up by the individual microphones in the handsets. It was tested by 120 participants in 10 different environments and allowed detecting the number of persons with an average error of 1.5 persons (Figure 2.26 - top left). No dedicated hardware is required to achieve this precision, enabling an application using off-the-shelf smartphones.

Microphones can also be used to analyze the mechanical surface waves that occur when objects interact with each other. Harrison et al. have designed TapSense, a microphone based sensor system that improves touch interaction by classifying the sounds created by different objects hitting the surface [HSH11]. In particular different parts of the hand can be distinguished, including nail, knuckle or tip. Potential applications are improving touch interaction on touch screens by enabling different forms of interaction, but also can be adapted to mobile devices, that typically have less interaction space and increase the expressiveness of different touch gestures. The achievable accuracy was between 95% using four different input types up to 99% when using just two input sets, such as finger and pen (Figure 2.26 - bottom).

2.3.5. Radiofrequency sensing

Radiofrequency sensing is a traditional field for sensors. Radar is a system that uses radio waves to acquire direction, speed, distance or altitude of objects and was developed at the beginning of World War II. This variety is using active sensing and emits radio waves that are reflected by objects. Most applications in smart environments similarly rely on active systems. A popular signal source is WiFi, intended to transmit information between several systems wireless. The systems are wide-spread, with most smartphones possessing two or more wireless technologies (typically 4G/3G/GSM for long range communication, WLAN for medium range communication and Bluetooth/NFC for short range communication) that can be used in coordination with stations placed in the environment. The signal processing of the wireless LAN sets generates some additional data that can be used, most notably the signal strength (RSSI). I will present different systems showing different ways, how radiofrequency sensing can be used in smart environments.

Radiofrequency based systems are very popular for indoor localization applications. I previously mentioned the difficulty to create time-of-flight systems in the electromagnetic spectrum, due to their high progression rate. Consequently, most systems rely on the received signal strength (RSSI). If the initial signal strength is known and we have a good estimate how the signal propagates, we can estimate the loss of signal strength at a certain distance. An important work that helped shaping this domain is the system created by Sugano et al. in 2006 that uses a ZigBee-based network with a limited number of nodes receiving RSSI information [SKOM06]. Based on this it is possible to locate one or more users with an error between 1.5 m and 2.0 m. This is often sufficient to distinguish the room a person is currently in, enabling a room-based system adaptation, which is suitable for many applications. A photo of a ZigBee node and results in a larger conference room are shown in Figure 2.27 on the left.

A different approach for RSSI based systems was presented by Wilson and Patwari in 2010 [WP10]. They are using tomography methods to determine the location of users. By placing a large number of sensors on the outer limits of the environment and creating a unique link between each, it is possible to infer the position based on the signal attenuation when a person moves in the environment. The human body absorbs some of the signal, resulting in a reduced received RSSI in the affected nodes. The error for standing persons was between 0.64 cm for a single person and 1.10 cm for two persons. An image of the test area and the resulting reconstructed attenuation map are shown in Figure 2.27 on the right.

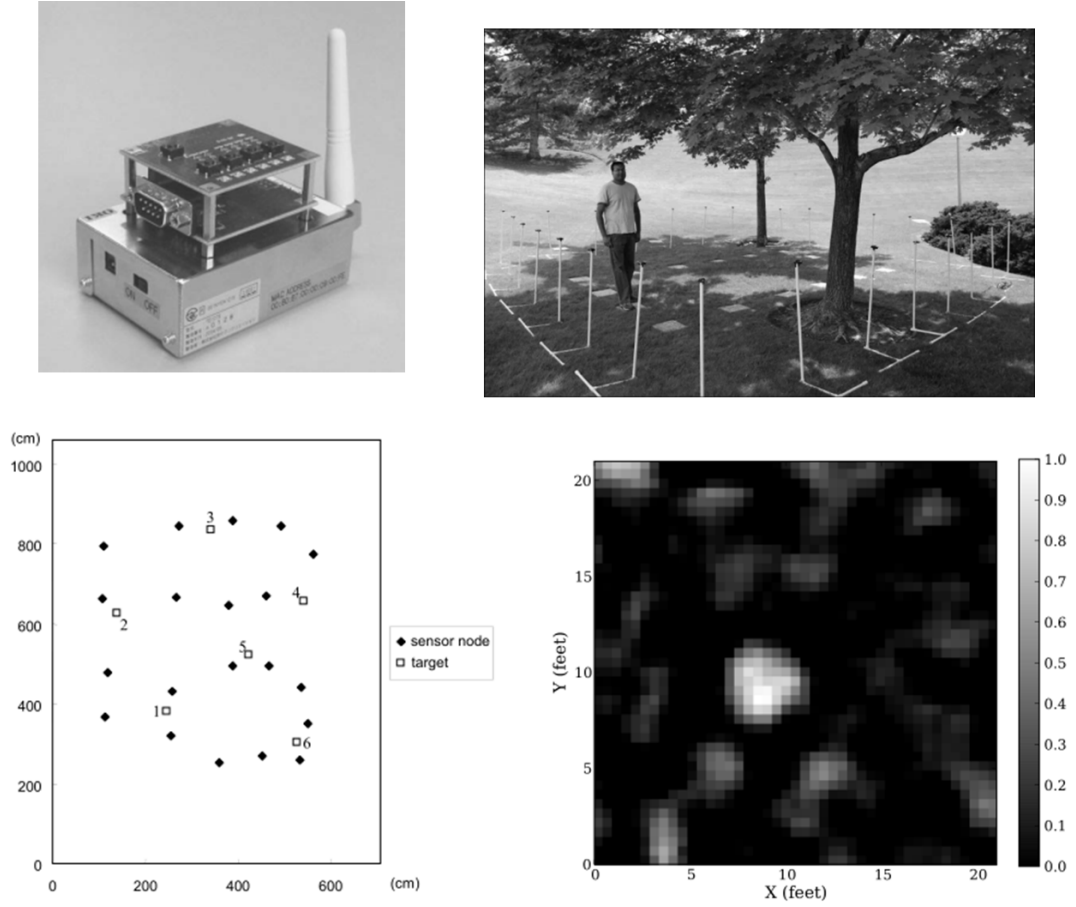


Figure 2.27.: *Top left* ZigBee node. *Bottom Left* Sensors and targets in larger room [SKOM06]. *Top right* Photo of radio tomography setup. *Bottom right* Result attenuated signal [WP10].

Another system that provides a localization is WiTrack, presented by Adib et al. in 2013 [AKKM14]. It uses the signal reflected by the human body to provide a location estimate based on time-of-flight. As mentioned previously this is challenging due to the propagation speed within the electromagnetic field. To overcome this issue they are using a method called frequency modulated carrier wave that transfers time differences to the frequency domain. Looking at the spectrum of received signal these shifts can be analyzed. The resulting location has an accuracy of 10cm to 13 cm in x and y and 21cm in z dimension. Additional use cases are provided in determining coarse arm or foot gestures and enabling an accurate detection of falls of a person. Some limitations of this approach include a restriction to just one person and that the tracked object has to move. In Figure 2.28 on the left we can see pictures of the antenna setup, the signal generator and how the location is determined in two and three dimensions.

The final radiofrequency system I want to present is WiSee, developed at the University of Washington by Pu et al. [PGGP13]. They are using WiFi to enable gesture recognition in large areas without requiring line-of-sight. They are analyzing the Doppler shift resulting from human activity in order to classify different gestures. Using

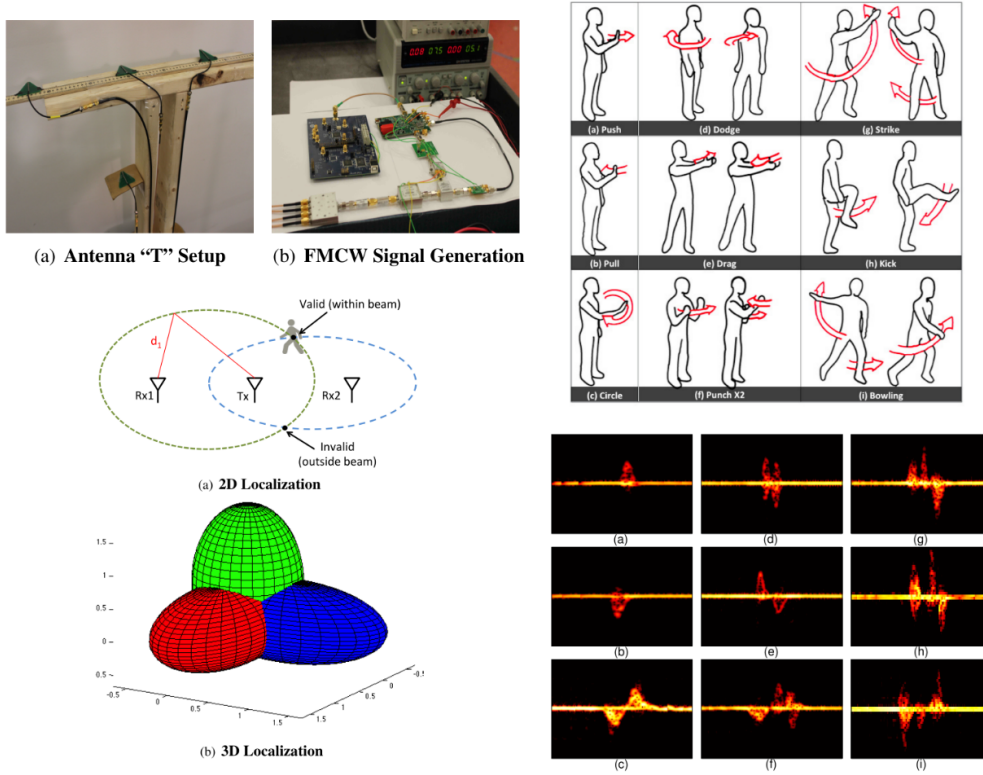


Figure 2.28.: *Top left* WiTrack antennas and signal generator. *Bottom Left* WiTrack 2D and 3D localization method [AKKM14]. *Top right* WiSee supported gestures. *Bottom right* WiSee Doppler profiles of gestures [PGGP13].

MIMO (multiple input multiple output) systems it is possible to distinguish between non-interacting persons in the environment and the person performing gestures. To equip a whole apartment only a single receiver and two transmitters were necessary. The achievable accuracy varies according to persons present and number/type of devices involved, but peaks at an accuracy of 94% for detecting nine different whole body gestures. The supported gestures and their Doppler profiles are shown in Figure 2.28 on the right.

2.4. Applications in smart environments

The field of smart environments is not strictly and conclusively distinguished from other fields in technology. It is using influences from disciplines including electric engineering, behavioral psychology, computer science, or mechanical engineering. Accordingly, it is difficult to formally list or distinguish all applications that are relevant or have been tackled in previous work. Thus, I will refer to previous collections of surveys, books and state-of-the-art in the associated disciplines smart environments, ambient intelligence and ubiquitous computing to get an informed selection of relevant applications that have been, or could potentially be supported by capacitive proximity sensors. The chosen collections of applications are taken from different collections of work that

Table 2.6.: Application domains and relevant works

Application Domain	Relevant Works
Indoor Localization	Braun - AmbiTrack camera system [BD13] Chintalapudi - EZ Indoor localization [CPP10] Mulloni - signpost markers [MWBS09] Pirkl - resonant magnetic coupling [PL13]
Gestural interaction	Zimmerman - hand gesture [ZLB*86] Wilson - XWand pointing interaction [WS03] Majewski - visual feedback [MBMK13] Liu - uWave [LZWV09] Pu - WiSee [PGGP13]
Physiological sensing	Cowie - emotion recognition [CDCT*01] Khosrowabadi - EEG emotion [KQWA10] Wöllmer - multimodal emotion recognition [WME*10] Hoque - MACH [HCM*13] Koelstra - DEAP database [KMS*12]
Activity recognition	Bao - user-annotated acceleration data [BI04] Bulling - electrooculography [BWGT11] Lasecki - real-time crowd labeling [LSKB13] Oreifej - Hon4D [OL13]
Smart appliances	Gellersen - MediaCups [GBK99] Yeo - StickEar [YNR13] Tsai - Smart Medication Dispenser [TCY*11] Dementyev - Bistable Display Tags [DGT*13]
Mobile devices	Ballages - mobile devices in ubiquitous computing [BBS06] Dearman - Bluetone [DT09] Vajk - TiltRacer [VCBE07] Nazari Shirejini - PECo [SN04] Olsson - mobile augmented reality [OKLVO12]
Autonomous systems	Coradeschi - symbiotic robotic systems [CS06] Broxvall - PEIS ecology [BGS*06] Arndt - robotic frameworks [AWD*13] Huijnen - companion robotics and smart homes [HBvdH*11]

have been released in the past few years. Cook et al. that presented a survey on recent developments in smart environments research in 2007 [CD07]. Augusto et al. edited a book on ambient intelligence in 2009, including chapters on various domains and applications. Another source is the book *Ubiquitous Computing* by Poslad, released in 2011 [Pos11]. Additionally, I will take into account recent conferences and journals that are active in this regard, such as CHI, UbiComp, Ambient Intelligent International, the Journal of Ambient Intelligence and Smart Environments, or the International Journal of Human-Computer Studies.

The selected areas-of-interest are collected in Table 2.6. I will briefly introduce relevant work in those domains and outline existing links to capacitive sensing applications.

2.4.1. Indoor localization

The domain of indoor localization has been briefly touched, when discussing the different sensing technologies used in smart environments in Section 2.3. The reliable tracking and localization of multiple users is a major challenge in smart environments. It is a very important contextual information that can be used to adapt the behavior of the environment. Often basic motion sensors are used, e.g. if just a single person should be detected or if there is a single area of interest that should be covered. However, determining a more exact location or following multiple users requires more sophisticated systems. If a varying number of non-relevant actors are moving in the environment, e.g. pets, the challenge is additionally increased. Together with Tim Dutz, I participated in the EvAAL competition 2013 with AmbiTrack, reaching a second place [BD13]. The system was based on a set of affordable off-the-shelf cameras connected to different processing nodes that employ an efficient tracking algorithm based on background subtraction, ray intersections and clustering methods [BDA*13]. It also supports analyzing the coverage of areas where it is installed, as shown in Figure 2.29.



Figure 2.29.: *Left*: Coverage of a room by camera fields of view - AmbiTrack [BDA*13]. *Middle*: Fiduciary marker photographed by camera phones [MWBS09]. *Right*: Magnetic coils in background and receiver circuit in foreground [PL13].

There are various indoor localization and tracking approaches that have been proposed in smart environments, including specific competitions that benchmark the different solutions against one another [CK12]. A common discrimination of localization systems is based on the type of object an actor has to wear in order to be successfully recognized. Active marker-based solutions require the person to carry a token that actively sends out signals. This is received by stations placed within the environment. Passive marker-based solutions require a token that is uniquely identifiable by external sensors, but is not sending any signal. Finally, there are marker-free solutions that do not require the actor to carry anything and instead rely on external sensors detecting unique features of the actor and following those within the environment.

A common method is to use radiofrequency sensing, tracking a mobile token, e.g. based on 802.11 WiFi networks. A recent work in this domain has been presented by Chintalapudi et al. [CPP10]. Their EZ system

relies on an existing WiFi infrastructure, the user carrying a mobile device with WiFi that also has periodic access to a GPS signal. It does not require any prior mapping or knowledge about the specific location or transmit power of the access points, but instead relies on a genetic algorithms to determine potential locations from a limited set of known locations and distances to access points calculated using the received signal strength (RSSI).

Another system was presented by Mulloni et al. [MWBS09]. They are using a set of fiduciary markers that are placed on signposts and can be photographed with smartphones. Based on an integrated mapping application the software will provide navigation from the current signpost to the desired destination. Thus, this is an example for an indoor localization system that relies on external markers, requires the user to wear a token in form of a smartphone and active user participation when the markers are photographed. The setup is shown in Figure 2.29 in the middle.

A final example is a system created by Pirkl and Lukowicz based on resonant magnetic coupling [PL13]. Using the magnetic field created by static coils, an arbitrary number of mobile receivers can localize themselves by measuring the field strength and estimating the distance from the different static coils. The system can be calibrated to ignore other magnetic objects in vicinity and enables a planar localization of about 1m^2 accuracy. The static coils can be seen in Figure 2.29 in the background with the receiver circuit in the foreground.

2.4.2. Gestural interaction

Gesture recognition enables the detection of meaningful expressions of motion by a human body, including the hands, arms, face, head and body [MA07]. If these expressions are translated into machine commands the result is gestural interaction. The most expressive and explicit form of gestures are performed by the hands, further distinguished into free-air gestures and touch gestures that typically involve one or more fingers interacting with a surface. The second variety is called multi-touch.



Figure 2.30.: *Left*: FTIR multitouch table [Han05]. *Middle*: DataGlove hand gesture system [ZLB*86]. *Right*: XWand interaction device [WS03].

Jeff Han showed a low-cost system based on frustrated total internal reflection (FTIR) of infrared light. This system allows tracking ten or more objects in real-time on large surface areas [Han05], as shown in Figure 2.30 on the left. Acoustic systems are another popular technology in this domain. Surface acoustic wave (SAW) uses the signal decrease of ultrasonic waves as they pass through an object touching the surface to infer its location [AAB98].

Throughout the years there have been various attempts to enable the tracking of gestures in free air. A common attempt in the late 1980s were data gloves, sensor-augmented gloves put on the hand that translated finger movements to gestural input, e.g. the system presented by Zimmerman [ZLB*86]. It uses optical sensors to measure the flex of the fingers and ultrasonic sensors to detect the absolute position and orientation of the hand. It is shown in Figure 2.30 in the middle. Applications included the evaluation of hand impairments and object manipulations in three-dimensional scenes.

A different approach is performing gestural interaction supported by interaction devices that can sense orientation and position in the room. A popular example is the Nintendo Wii Remote that is used in gaming applications. A predecessor was the XWand by Wilson et al. [WS03]. This interaction device is using accelerometers to sense orientation and has infrared diodes that are tracked by an external camera system to determine the absolute position in the room. It is shown in Figure 2.30 on the right. Using knowledge about the location of different appliances within this room, it is possible to control them by pointing in this direction. They later extended the system with a pointing device based on a laser that indicated the position in the environment currently pointed at [WP03]. We have used a similar method, based on skeleton tracking provided by a Kinect and different varieties of pointing gestures [MBMK13].

An extension of accelerometer-based gesture systems was presented by Lui et al. [LZWV09]. Their uWave system is capable of building a personalized gesture set using dynamic time warping and HMM classification, enabling a high recognition rate of almost 99% using just one training sample. Additional applications of this approach include using personal gestures for authentication, reaching an error rate of approximately 3%. However, this method can be bypassed if the movements are observed and mimicked and thus is not suitable for critical applications.

I previously introduced WiSee presented by Pu et al. [PGGP13]. Using two sources of wireless signals, they are using the Doppler shift caused by the human body moving in the area and reflecting the signal to determine gestures. The system was able to detect nine different full-body gestures with a precision of 94%. The system uses the MIMO capabilities of modern WiFi systems to distinguish users and providing the option to support multiple users within an environment.

2.4.3. Physiological sensing

In the 1990s researchers began to distinguish different channels when interacting with machines. The explicit channel, whereas the user gives distinct commands to the user, and the implicit channel that comprises information about the user himself [CDCT*01]. One interesting parameter to consider for this implicit interaction is emotion. Based on physiological cues it is possible to determine the current state of the user and translate it to an input for machines. Thus, an area has emerged that uses acquired physiological signals of the users to provide additional input. In this section I will present four different works of recent years that proposed important technologies and methods.

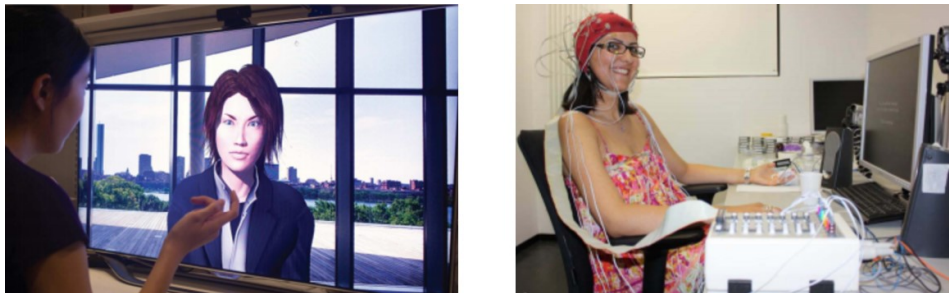


Figure 2.31.: *Left*: MACH conversation coach system [HCM*13]. *Right*: Study setup to collect emotion data [KMS*12].

Khosrowabadi et al. presented a machine learning approach to discriminate four different emotions from EEG readings [KQWA10]. The subjects were asked to fill out self-assessment questionnaires regarding their emotional

state with the states being induced by a set of pictures. The signals were classified using a self-organizing map and nearest-neighbor classification. The system was able to correctly classify 84.5% of four different emotional states.

A multimodal emotion recognition system based on speech and facial expression was introduced by Wöllmer et al. [WME*10]. They apply a method based on long short-term memory networks that allows to model long-range temporal dependencies between features. They combine 46 facial markers and a variety of waveform speech features that are combined using a correlation-based feature subset selection that leads to a selection between 66 and 224 features. They are able to improve the recognition performance compared to HMM-based classifiers by 4%.

An example application for emotion-aware systems has been presented by Hoque et al. [HCM*13]. *MACH: My Automated Conversation coach* is a training system that tries to improve the social skills of its users. It collects facial expressions, speech data and analysis prosody from an attached camera to create a personalized feedback of the user's behavior when talking to a virtual agent. This agent is shown in Figure 2.31 on the left. A study with 90 participants showed a significant performance improvement as opposed to the control group.

In this domain it is also very important to build databases of labeled emotions and physiological signals that allows other researchers to try new methods. Koelstra et al. [KMS*12] created a database by collecting EEG information, playing different video clips and collecting the perceived emotions in a self-assessment. This database includes the physiological signals of 32 users and frontal face videos of 22 users that are reacting to 40 different videos. The setup of this study can be seen in Figure 2.31 on the right.

2.4.4. Activity recognition

Activity recognition, also called situation awareness, describes the interpretation of sensor data into higher domain-level information that relates to the given situation [YDM12]. To give an example, a single temperature reading of 15°C can be combined with the information that the sensor is in the sleeping room, the time is 21:00 and we know that the inhabitant of the environment is typically going to sleep at 22:00. In this case we can infer the activity *user will go to bed soon* and trigger the action *heat up sleeping room*. There is a plethora of different concepts and methods on how to recognize the situations and activities for numerous domains. A good overview of the topic can be found in the survey of Ye et al. [YDM12]. I will present four different works that have been presented in recent years that provide a different selection of technologies and methods that are commonly used.

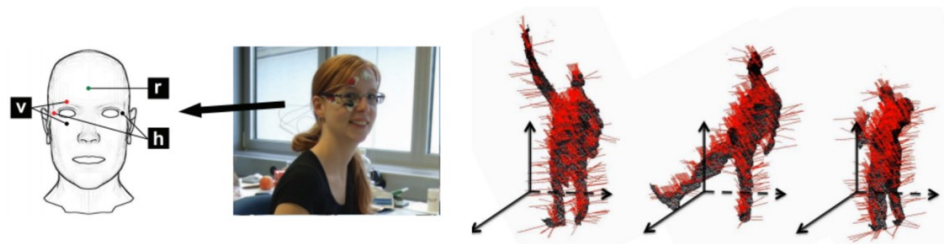


Figure 2.32.: *Left*: Eye movement tracker by Bulling [BWGT11]. *Right*: Hon4D - surface normals of different activities [OL13].

When trying to determine human activities wearable sensors are an important approach that has been used extensively. It is possible to use either physiological measurements or movement data. A classic work in this

domain presented by Bao et al. is associating activities based on movement data generated by accelerometers [BI04]. This sensor category is able to detect the acceleration in multiple directions, e.g. commonly used to detect the orientation of mobile devices. Using five different sensors attached to arms, legs and hip, they were able to associate 20 different activities performed by 20 different subjects with an accuracy of approximately 80%.

In some application domains that have a specific set of tasks other approaches might be suitable. An interesting system was presented by Bulling et al. that tries to determine activities based on the movement of the eyes [BWGT11]. They are attaching a set of electrodes to the user's head to track the activity of muscles around the eye, without using any external sensors such as gaze trackers. The setup and electrode placing is shown in Figure 2.32 on the left. In a study with eight users they tried to distinguish five different typical office work activities using SVM classification. These activities are copying text between documents in a two monitor setup, reading a document on the table, writing on a page on the table, watching a video and browsing the internet. The achieved average precision was 76% and recall 71%.

A common problem in situation classification is labeling of the acquired data to a given situation. Typically this is performed manually by the researcher. Lasecki et al. introduced Legion:AR, a system that allows to crowd-source this process to a large group of workers that individually label activities from a video feed [LSKB13]. They could show that while in complex situations a group of five persons labeled 90% of activities and objects correctly, while a single person only reaches 48% on average. The process is aided by integrating the input from different persons that are currently labeling.

A final example in the domain of activity recognition is Hon4D, a system that infers situation from depth camera sequences [OL13]. They suggest the histogram of normal orientation in depth, time and spatial coordinates as a feature for activities from depth data. Examples of surface normals for different activities and body surfaces are shown in Figure 2.32 on the right. The advantage of this operator is that it takes into account the movement of the overall surface of recognized persons, as opposed to silhouettes or the reconstructed skeleton joints. This approach reaches classification precision between 89% and 97% when used on common activity depth data sets.

2.4.5. Smart appliances

Smart appliances are devices that are attentive to their environment [SV01]. This is usually achieved by integrating different sensors and actuators to provide additional functions and services to a user. Some examples include intelligent furniture that can detect their occupation, internet-connected household items, or single-purpose devices, e.g. providing reminder services. An overview of different examples can be found by Park et al. [PWLK03].



Figure 2.33.: *Left:* Different MediaCup prototype [GBK99]. *Right:* Stickear system to augment appliances with audio sensing [YNR13].

One example for common household items augmented with additional features is the MediaCup by Gellersen et al. [GBK99]. This coffee cup is augmented with temperature and acceleration sensors, a processing unit and communication using an infrared system. Some of the prototypes are shown in Figure 2.33 on the left. It is able to sense if there is fresh coffee in the cup, if it currently used for drinking, stationary or being played with. The applications were focused on remote colleague awareness, whereas the activities of the MediaCup were transmitted to a remote location.

StickEar by Yeo et al. is a wireless device that adds different capabilities to objects it is attached to [YNR13]. They integrate a rotary switch, buttons, microphone, speaker and processing and communication components into a small portable package, as shown in Figure 2.33 on the right. Some supported interactions are control of devices using sound, autonomous response to sound events, or remote triggering of sound. The system can be controlled using an app for mobile devices.

There is a number of smart appliances in the medical domain. They try to provide different services related to health and well-being. A common example are smart medication dispensers, such as the one presented by Tsai et al. [TCY*11]. Based on a medication schedule it will dispense the right medication and dose at the specified times. They include a few different algorithms for heuristic scheduling based on a collaborative approach between scheduler and the controller of the dispenser.

When deploying smart appliances it is necessary to provide energy to the systems. If there is no socket nearby batteries or other methods have to be used. If the system is capable to independently harvest the energy from environment sources. One example are the bistable display tags by Dementyev et al. [DGT*13]. They rely on e-paper displays that keep their screen content unless a refresh is triggered. Using an energy harvesting IC it is possible to transfer enough information and the new screen content via NFC. In an example application screenshots from a smartphone could be transferred to the display.

2.4.6. Mobile devices

In his famous article, Mark Weiser coined three different forms of devices - tags, tabs and boards [Wei91]. They are primary distinguished by their size and how they can be used for interaction with the system. The prevalent smart phones and tablets nowadays resemble closely the envisioned smart tab - handheld devices that provide sensing, interaction and communication facilities supported by sufficient processing power. Consequently, they are used very often in smart environment applications.

Ballages et al. collected and investigated numerous ways how smart phones can be used to in ubiquitous computing [BBRS06]. They provided an overview of different positioning, orientation and selection methods, including using the camera on the mobile phone and different processing methods to control a cursor. An important contribution was a classification of potential mobile phone interactions and a summary of position, orientation and selection techniques.

Blutone is a system created by Dearman and Truong that allows to control public displays using mobile phones via Bluetooth communication [DT09]. In order to avoid installing any additional application they are using the Bluetooth headset profile and transfer sounds instead of packaged information. Similarly Vajk et al. proposed using the accelerometers in a phone to control applications on a large public display [VCBE07]. One example application was TiltRacer that could be controlled by different users standing in front of a display, as shown in Figure 2.34 on the left.

Nazari Shirejini presented PECO, an environment controller based on a PDA device [SN04]. Based on a 3D model of the current environment it was possible to control different appliances by selecting them on the mobile device. Additionally, a concept was presented that allows to transfer documents from the PDA to different suitable devices, by using simple drag & drop operations. The practical use case was implemented in a lecture

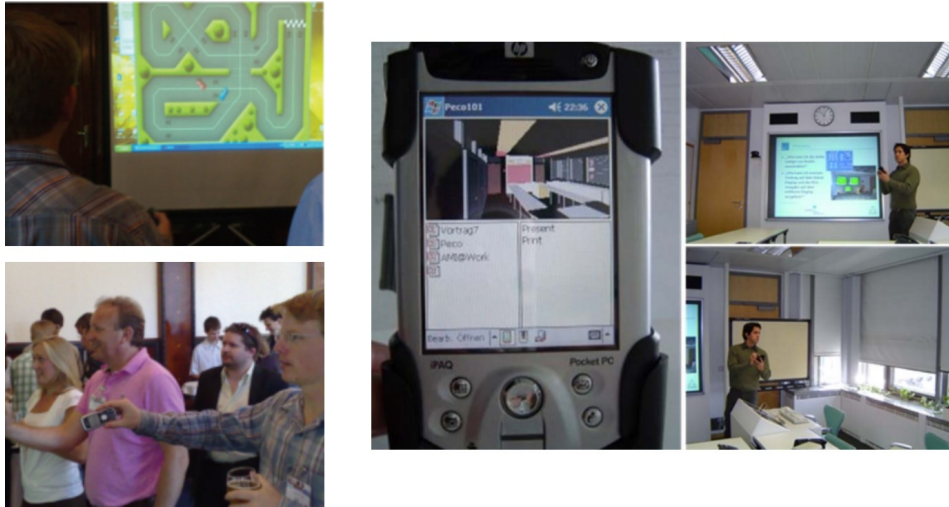


Figure 2.34.: *Left*: TiltRacer controlled by accelerometers of mobile phones [VCBE07]. *Right*: PECO environment control application on a PDA [SN04].

room and connected to the controlling system, e.g. allowing to display documents on a projector by dropping them on the 3D model of the projecting screen. The system is shown in Figure 2.34 on the right.

A final example is in the popular domain of augmented reality systems for mobile devices. There are numerous popular applications that overlay additional information on a live camera image of mobile phones. Olsson et al. tried to evaluate user experience and acceptance of different mobile augmented reality applications [OKLVO12]. They found that information applications were better received than entertainment applications. In addition, information flood, loss of autonomy and virtual replacement of actual items were seen as most negative aspects of this technology.

2.4.7. Autonomous systems

Autonomous systems are an important future application in smart environments. While factory complexes started using robots decades ago, the trend towards home robotics is fairly recent, as the processing and sensing capabilities of the systems increased, while the price could be reduced significantly. There are numerous potential use cases, ranging from vacuuming or gardening robots, to full-service robots that can be used in care systems for the elderly. In this section I will present four different examples, how autonomous systems can be used in smart environments.

Coradschi and Saffiotti propagated the symbiotic properties of robots operating in a smart environment [CS06]. Human, robot and smart environment are modeled as three distinct actors that share resources and information between each other. The distributed systems in the environment, e.g. sensors, tags and actuators, can deliver additional information to the control unit of the autonomous system that can be used to optimize any strategies. Similarly, users can benefit from the coordination between robot and environment to achieve common goals.

An implementation of a similar concept was published by Broxvall et al. [BGS*06]. PEIS, a network of heterogeneous smart devices that operate in a smart environment. In the example scenario shown in Figure 2.35 on the left. A coordinating system (Emil) sends commands to a robot (Pippi) that moves around the house but has

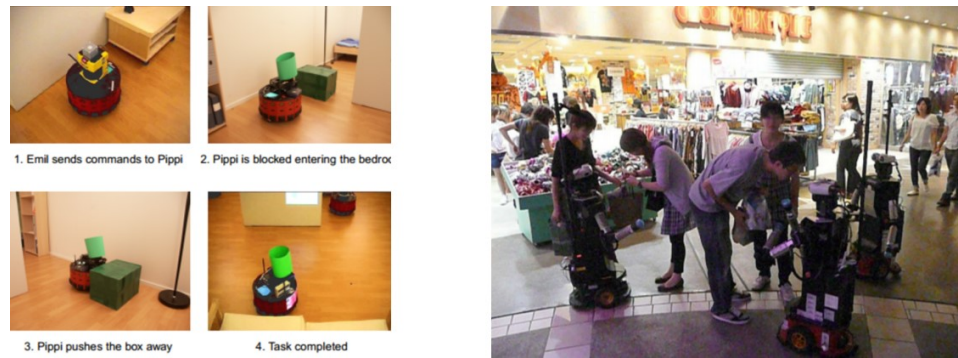


Figure 2.35.: *Left*: Scenario of the PEIS system [BGS*06]. *Right*: Robots moving around in a social environment [GKIH09].

his path blocked by a parcel. As this parcel is a smart object and knows its contents this information is forwarded to the robot, who then safely moves the parcel away. Thus, a collaboration between smart environments and robots can support similar scenarios that reduces the requirements for both environments and robots.

Glas et al. investigated how robots can track people and localize themselves in social environments that are frequented by a larger number of persons [GKIH09]. They combine laser range-scanners placed in the environment that provide wide coverage and are able to track the trajectories of moving persons and robots. This is combined using Kalman filtering with the odometry data generated by the robots. As a combined solution this enables localization of both people and robots in a shared global coordinate system. One example application system is shown in Figure 2.35 on the right, whereas robots provide directional information to shoppers in a mall.

A final example that evaluates user experiences gathered in different research projects was presented by Huijnen et al. [HBvdH*11]. Within the projects CompanionAble and Mobiserv different service robots were deployed in smart home environments for elderly persons. The user trials involved testing interaction and usability, user acceptance and privacy. The robots user - Hector and Kompai - are designed to provide assistance for persons with Mild Cognitive Impairments or early stages of dementia. One result was that the appearance of the robot was not particularly important for this user group, as opposed to the functionality, which was considered most important.

2.5. Disucssion

In this chapter I have given an introduction to electric field sensing, including its physical properties, the influence of layout and materials, and an outline of different processing methods. I have shown a number of applications built upon this technology, ranging from historic examples to research done in the last years. In order to compare capacitive proximity sensors to competing technologies, I have introduced several multi-purpose sensor technologies that are frequently used in applications for smart environments. Additionally, I have given an overview of application domains within smart environments. These are considered as candidates for systems based on capacitive proximity sensors. These related works motivate the benchmarking model that will be presented in the next chapter. It is necessary to have an extensive overview of potential applications and sensor technologies to derive a benchmarking model that successfully links sensor capabilities to features required for certain applications.

3. Benchmarking model for sensors in smart environments

When designing a new application or system for a specific purpose, the parties involved have to make a number of decisions regarding the different components, processes and methods that are to be used. Benchmarking is a method that is mostly used in business practice, in order to compare the performance of processes, products and market entities against one another. Either one or multiple different indicators are used to act as a metric or take part in the calculation of an overarching metric that can be used to compare performance relative to other entities [CC89]. This tool is widely used for supporting decisions in different domains. Looking at smart environments, a common challenge is to select a specific sensor technology for any given application. The majority of systems are following a structured approach in the design process, e.g. by ranking available systems or performing an iterative trial and error routine. However, so far there has been no generic model that would allow to evaluate the expected performance of a system based on a specific sensor technology. Using a set of common features and an adaptive weighting model, it is possible to cover a high number of different applications in a specific domain and thus support the decision process at an early stage of the application design.

In this chapter I will introduce a formal benchmarking model that allows estimating the performance of applications in smart environments based on a specific sensor technology. To build a basis for this modeling it is necessary to look into previous works related to technology benchmarking and the selection of suitable metrics. Additionally, it is necessary to find a common set of features relevant for smart environments that can be applied to a number of different sensor technologies. The method is evaluated by benchmarking several example technologies and applications based on querying large scientific databases. Finally, the benchmarking method is used to determine different applications suitable for capacitive proximity sensors. This chapter is extended from [BWK14].

3.1. Benchmarking

Benchmarking is a tool that is widely used in computing technology [LC85]. Hardware benchmarks compare the performance of different single systems, often seen for GPUs or CPUs to evaluate both theoretical and real-life performance. Some metrics that are used for theoretical comparison in CPUs are FLOPS (floating point operations per second), e.g. measured by Linpack [DLP03], or MIPS (million instructions per second), e.g. measured by Dhrystone [Wei84]. Regarding GPUs the benchmarks include Texel rate (how many triangles can be processed per second) and Pixel rate (how many pixels are processed per second). Real-life benchmarks for CPUs typically included timing specific tasks on applications that are demanding for certain aspects of the CPU, such as video processing, image processing or audio encoding. For GPUs many PC games provide benchmarking tools that allow evaluating the real-life performance of different graphics cards at different settings, including modifications to resolution or detail level. The typical metric here are FPS (frames per second) that denote how often the screen content can be rendered in a second.

System benchmarks are a step up from single component benchmarks and combine the performance measurements of various components in different scenarios to evaluate the estimated behavior in numerous real-life situations. There are several standardized test suites that provide this functionality, such as SPEC [Hen00]. A common single index that is available for all newer Windows machines (Vista and beyond) is the Windows System Assessment tool that calculates the WEI (Windows Experience Index), a combined score of CPU performance, 2D and 3D graphics performance, memory performance and disk performance [Whi06]. The lowest score of all single metrics is chosen to determine a lower bound for expected real-life performance. Ranganathan et al. introduced benchmarking methods and a set for pervasive computing systems [RAMB*05]. They distinguish system metrics, configurability and programmability metrics and human usability metrics. Finally, if different systems of the same category are compared, technology reviewers often use a single index that is calculated based on various aspects of the system. Smith introduced different potential combined metrics that can be used for this purpose [Smi88]. Three different approaches are outlined - geometric mean, arithmetic mean and harmonic mean. Additionally, varieties with a specific weighting are mentioned. Another example for benchmarking whole systems is the EvAAL competition that aims at evaluating different technologies that are applicable in Ambient Assisted Living [CK12]. There are various tracks, including indoor localization and activity recognition. Apart from technical metrics, such as precision, a focus of this competition is on a more holistic approach and thus includes metrics like installation time, user acceptance and interoperability of the solution. Finally there has been considerable work in the domain of identifying suitable metrics for a given benchmark. Crolotte argued that the only valid benchmark for decision support systems is the arithmetic mean of different single benchmark streams, as it is valid for normalized and time-relevant benchmarks [Cro09]. Jain dedicates several chapters of his book to introduce methods and considerations for metric selection in benchmarking computer systems [Jai91].

3.2. Sensor features

One of the most challenging aspects of benchmarking is selecting the appropriate metrics to be included in the scoring process. In order to identify relevant sensor features for technologies to be applied in smart environments, inspiration is taken from sensor technology overviews [Wil04] and the pervasive model presented by Ranganathan et al. [RAMB*05]. Accordingly, it is possible to identify three different groups of sensor features: sensor performance characteristics, pervasive metrics and environmental influence. These different groups are detailed in the following sections. An overview of the different potential members of this groups is given, their relevancy for the benchmarking model is discussed, and a feature matrix is created that builds the basis for the feature scoring model.

3.2.1. Sensor performance characteristics

This group of sensor features is related to specific technical properties of the given sensing device, as they would be typically put into the datasheet. A first important characteristic is the sensitivity or resolution of a sensor, which is the smallest change of a measured quantity that is still detectable. For example an accelerometer might be able to only detect changes that are above 0.1g. Another important characteristic is the update rate of a sensor. This denotes the number of samples the sensor is able to measure in a certain timeframe. Typically, the number of samples in a second is noted as frequency, thus a sensor may have an update rate of 20Hz, generating 20 samples in a second. Another factor that is particularly important for embedded systems or wearables is the power consumption of the sensor that may limit the time it can operate on battery, independent of a power source. A last example is the detection range, denoting the maximum distance between the quantity to be measured and the sensing device.

3.2.2. Pervasive metrics

Pervasive metrics can be identified as features that specify how well a given sensor system will perform in collaboration with smart environments, when networked with other devices, or when placed into existing surroundings. An example for the latter is the obtrusiveness of a sensor device. If it is clearly visible when applied, if there are disturbing signals generated, or if certain privacy concerns are associated to the sensor device, the acceptance by the user and thus the applicability is reduced. If the sensor is operating in a larger network of other devices, the bandwidth required to submit the signal to an analyzing node should be kept low. Equally, if the processing capabilities are limited, less complex data processing is preferable. The overall system cost is increasing if single sensors are particularly expensive, thus limiting the potential applications. The system and attached sensors should be robust, both in terms of physical design and quality of service. Finally, the sensors are more readily applicable if the systems are interoperable to each other.

3.2.3. Environmental characteristics

The third group is the environmental characteristics of a sensor system. Any sensor is affected by a certain disturbance caused by factors in the environment that are similar to the measured quantity, also called noise. For example an optical sensor is influenced by ambient light sources. In this context it is relevant how frequent those influences are in a certain environment and how robust the sensor is against noise. In many cases the presence of noise can be detected and counteracted with a calibration towards the changed environmental factors. The complexity of this calibration is another interesting factor in this regard. Finally, all sensors have some unique limitations, e.g. specific materials that absorb certain wavelengths of the electromagnetic field are difficult to detect for sensors that work in this specific frequency range.

3.2.4. Discussion and feature matrix

To keep the benchmarking constrained, the three most relevant features of each category are chosen. This allows a more manageable overall model. However, it requires an informed selection of the presented features. Of the sensor performance characteristics group resolution, update rate and detection range are selected. Resolution is a critical feature in any application, determining precise any detection is and if particular objects may be detected at all. Update rate is equally important if fast objects are to be detected and if a reactive systems is desired that respond in real-time. The importance of detection range correlates with the size of the environment and may lead to a reduction of required sensors. Of the mentioned features power consumption is omitted. The actual power consumption of a whole system is a more interesting metric but very difficult to predict from the energy usage of a single sensor [LWG05]. Of the pervasive metrics group unobtrusiveness, processing complexity and robustness are chosen. Unobtrusiveness of the sensor device is a desired feature in many different scenarios, where it should not impede the environment. While microprocessors are becoming ever faster processing complexity is still crucial if the number of sensors is increasing. A dedicated chip will require a more complex architecture and lead to more cost, higher energy usage and more potential points of failure, leading to the final chosen feature of robustness, both against physical abuse, but also in terms of system design, where it is supposed to be resilient towards failure of single components.

Additionally, the required bandwidth was omitted, as this metric is not important for many sensors, as they have low bandwidth requirements in general. In the last years the available bandwidth in wired and wireless systems is increasing continuously. In the last group of environmental characteristics frequency of the disturbing factor, calibration complexity and unique limitations are chosen. If the disturbing factor occurs only rarely, it is not critical and therefore can lead to favor a specific technology in the benchmark. The calibration complexity

3. Benchmarking model for sensors in smart environments

Table 3.1.: Feature matrix denoting capabilities required for a certain rating

Feature	–	-	o	+	++
Resolution	very coarse	coarse	normal	fine	very fine
Update Rate	less than once per second	slower real-time	real-time	faster real-time	more than 100 times per second
Detection Range	touch	less than one meter	less than 5 meters	less than 20 meters	more than 20 meters
Unobtrusiveness	open large system	open small system	hidden system, large exposure	hidden system, small exposure	invisible
Processing Complexity	single sensor CPU	10+ sensors CPU	single sensor embedded chip	10+ sensors single embedded chip	no further processing
Robustness	single point of failure	error detection	quality of service	self-recovery	fully redundant
Disturbance Frequency	very frequent	frequent	average	unlikely	highly unlikely
Calibration Complexity	very hard	hard	normal	easy	very easy
Unique Limitations	very critical	critical	average	not critical	none

combines both the processing complexity and time that is required to recalibrate the system. This is highly important in real-time systems that have to monitor the environment continuously. Finally, unique limitations are a rather broad metric that is difficult to quantify. However, in many scenarios it is obvious that a specific limitation might arise, e.g. if the smart environment is in an area with a lot of human noise, microphones could be regularly disturbed. Including this metric allows modeling those applications into the benchmark with a strong weight penalizing unsuited sensors. From the selected metrics a feature matrix is created that allows to associate specific capabilities to a specific rating that will be used later in the scoring process of the benchmark model. Each feature is mapped to five different ratings on an ordinal rating scale comprised of the items “least favorable” (–), “not favorable” (-), “average” (o), “favorable” (+) and “very favorable” (++). This leads to the feature matrix shown in Table 3.1, which will be discussed briefly.

- *Resolution* is ranging from “very coarse” to “very fine”. This unspecific rating is used, as the range may vary strongly between different sensor types. A mapping to actual should depend on the application and object that has to be detected. If the object is large a sensor that would be ranked “coarse” for smaller objects can be ranked as “fine”.
- *Update Rate* is rated around real-time performance that is often rated at around 20 samples per second. Slower sensors might miss various events, while faster sensors allow detecting highly dynamic events.
- *Detection Range* is rated around the 5m distance mark, that is typically enough to cover the entirety of a single apartment room. For larger rooms sensors with a higher detection distance are favorable, many sensors only react to touch.
- *Unobtrusiveness* is ranging from exposed systems placed in the environment (one example would be the Microsoft Kinect) to invisible systems that integrate seamlessly into the environment.

- *Processing Complexity* has a range from dedicated CPUs that are required to process the data of a single sensor to smart sensors that require no further processing, which allows to apply numerous sensors without adding additional processing capabilities to the environment.
- *Robustness* is following criteria for quality of service. The least favorable system fails, if only a single node is present and failing. The preferred system is fully redundant.
- *Disturbance Frequency* is ranging from frequently occurring disturbing signals, to highly unlikely disturbing signals, resulting in a better rating.
- *Calibration Complexity* is a combined metric including the calibration time, the required processing capabilities and if external aid is required in the calibration process, leading to a rating from “very hard” to “very easy”.
- *Unique Limitations* should be ranked according to their criticality, as previously explained they may be suitable to penalize certain sensors or emphasize the prevalence of a disturbing factor in a noisy environment.

Now that the feature matrix is complete, the next step is presenting the formalized benchmarking model and how to use the presented features and their rating to calculate a benchmark score that allows comparing different sensor categories with regard to different applications.

3.3. Benchmarking model

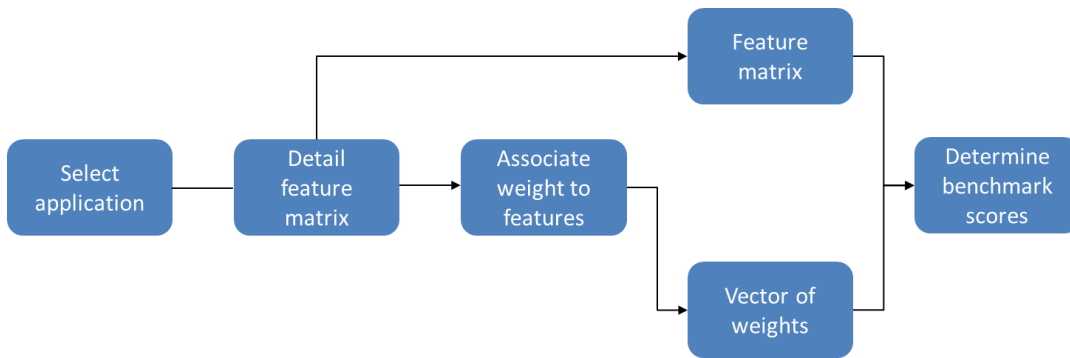


Figure 3.1.: Benchmarking process

In this section I will describe a formal model that will allow us to determine a benchmark score for a given application and a given sensor technology. As previously explained the different applications are distinguished by applying a different set of weights to the known features. As a start the process of this feature weighting is discussed and some examples about proper application are given. Afterwards, the formal model is introduced that deduces a single score benchmark for any sensor technology and any application. The overall process is shown in Figure 3.1 and will be detailed in the following sections, including some example calculations.

3.3.1. Feature score and weighting

The presented feature matrix has some ratings that need detailing in order to be quantifiable in the specific application. The ordinal measurements of the feature matrix should be assigned a quantifiable measure. Regarding

“Unobtrusiveness”, the open system can be detailed as “visible by users” and “large system” as size larger than 100mm x 100mm x 100mm. Similar levels of detail can be applied to the other features leading to the application-specific detailed feature matrix that is used in the scoring process. The different ratings are assigned different numeric values, namely 0.00 (–), 0.25 (–), 0.5 (o), 0.75 (+) and 1.00 (++). The weight of the features for the specific application is also rated on a 5-point ordinal scale, denoted as “not important” (numeric value 0.0), “less important” (0.25), “moderately important” (0.5), “important” (0.75) and “very important” (1.00). Thus, for each application there is a distinct detailed feature matrix and a vector of associated weights that can be applied to a set of sensor technologies, in order to calculate the benchmark score. The next step is to formally introduce the model that allows to determine a benchmarking score from feature ratings and application weights.

3.3.2. Modeling

The model is supposed to formalize a benchmark for any application and any sensor technology in any domain. As a start the following definitions are given:

- Set of n domains $D = \{d_1, \dots, d_i, \dots, d_n\}$
- Set of m applications $A = \{a_1, \dots, a_j, \dots, a_m\}$
- Set of o features $F = \{f_1, \dots, f_k, \dots, f_o\}$
- Set of p sensor technologies $S = \{s_1, \dots, s_l, \dots, s_p\}$

In any domain d_i there is have a set of potential applications $A_{d_i} \subseteq A$ and a set of relevant features $F_{d_i} \subseteq F$. For each feature $f_k, d_i \subseteq A$ there is the associated feature score $r_{F_{d_i}}$ as explained in the previous section. Each sensor technology s_l has a specific feature score $r_{s_l, F_{d_i}}$ with normalization $\|r_{s_l, F_{d_i}}\| \in [0, 1]$.

$$\overrightarrow{r_{s_l, F_{d_i}}} = \begin{pmatrix} r_{s_l, f_{1,i}} \\ \vdots \\ r_{s_l, f_{o,i}} \end{pmatrix} \quad (3.1)$$

The weights w_{f_o} associated to a specific application a_j in a domain d_i have the same cardinality $|w|$ as the vector of feature scores $\|\overrightarrow{r_{s_l, F_{d_i}}}\|$. The values are determined, so $\|w_{f_o}\| \in [0, 1]$.

$$\overrightarrow{w_{a_j}} = \begin{pmatrix} w_{f_1, a_j} \\ \vdots \\ w_{f_o, a_j} \end{pmatrix} \quad (3.2)$$

The feature scores and associated weights allow to determine a benchmark score b_{s_l} for a specific sensor technology s_l for any application a_j by using the scalar product of feature score and respective weight and apply normalization with regard to the weight.

$$b_{s_l} = \frac{\overrightarrow{r_{s_l, F_{d_i}}} \cdot \overrightarrow{w_{a_j}}}{\sum_{k=1}^o w_{f_k, a_j}} \quad (3.3)$$

Thus it is possible to compare different sensor technologies by calculating and comparing the different benchmark scores for a given set of sensor technologies $S_p \subseteq S$ and receive a set B_{S_p} with $t = |s_p|$.

$$B_{S_p} = \{b_{s_l, 1}, \dots, b_{s_l, t}\} \quad (3.4)$$

In order to determine the optimal (chosen) sensor technology b_c for an application a_j and given the prerequisites regarding non-negativity of weights and feature scores we can evaluate the set for the maximum element.

$$b_c = \max(B_{S_p}) \quad (3.5)$$

3.3.3. Feature score normalization

With regards to actual benchmarking the problem of bias towards a specific technology may occur. If the average features ratings are different between two technologies the calculated benchmark score will increase. In many instances this might be beneficial, yet if comparing numerous technologies to a set of different applications a trend might be more important than absolute scores. Thus, an optional step is provided that calculates the normalized feature vector $r_{s_l, F_{d_i}, norm}$ with regard to the average associated value of 0.5, using the following equation.

$$\overrightarrow{r_{s_l, F_{d_i}, norm}} = \begin{pmatrix} r_{s_l, f_{1,i}} \\ \vdots \\ r_{s_l, f_{o,i}} \end{pmatrix} \cdot \frac{o \cdot 0.5}{\sum_{p=1}^o r_{s_l, f_{p,d_i}}} \quad (3.6)$$

The feature-normalized benchmark score is accordingly determined with the following equation.

$$b_{s_l, norm} = \frac{\overrightarrow{r_{s_l, F_{d_i}, norm}} \cdot \overrightarrow{w_{a_j}}}{\sum_{k=1}^o w_{f_k, a_j}} \quad (3.7)$$

3.3.4. Scoring

Now with the formal model and the available set of feature matrix and weights it is possible to calculate the benchmarking score for a set of sensor technologies. As an example the application indoor localization in a public shopping area is chosen. A typical use case would be the monitoring of customer behavior. The requirements include a tracking accuracy of 50 cm, with a large area to cover and potentially fast moving persons. Thus, the importance ratings for performance characteristics are moderately important for resolution, important for update rate and very important for detection range. As the system can also be used for security purposes, the unobtrusiveness is less important. There can be dedicated servers, so processing complexity is not an issue, but the system should be difficult to disturb, thus robustness is considered important. Disturbance frequency is moderately important, as a large number of persons is monitored, leading to statistically significant results, even if single measurements are disturbed. The environment is fairly static, thus calibration complexity is less important. It is possible that a crowded shop produces a lot of acoustic noise, therefore no unique limitations towards acoustic disturbances should be present and this is moderately important. The resulting vector of weights is:

$$\overrightarrow{w_a} = (0.50 \quad 0.75 \quad 1.00 \quad 0.25 \quad 0.00 \quad 0.75 \quad 0.50 \quad 0.25 \quad 0.50)^T \quad (3.8)$$

Based on previous experiences and best practice a camera-based system is evaluated. The system has high resolution cameras, with an update rate of 30 samples per second and a high detection range of more than 20 meters. The cameras are external, not hidden from view but attached on the ceiling. The processing complexity is very high, requiring a dedicated system per camera. Since they are out of reach they are robust towards human intervention and independent from each other. In the given setting visual disturbance is unlikely, calibration is difficult but not required regularly and the system is not disturbed by acoustic noise. This results in the following rating vector:

$$\overrightarrow{r_{s,f}} = (1.00 \quad 0.75 \quad 1.00 \quad 0.25 \quad 0.00 \quad 0.50 \quad 0.75 \quad 0.25 \quad 1.00)^T \quad (3.9)$$

Using those two vectors the final scoring for this sensor system can be calculated, using the equations of the previous section, leading to $b_s \approx 0.78$ and a feature-normalized score of $b_{s,norm} \approx 0.63$. Determining the feature rating vector for other technologies is possible in a similar fashion. The optimal technology would have the highest score b_s or $b_{s,norm}$.

3.4. Evaluating technology popularity

In order to evaluate the method, I propose a verification based on previous successful works in the domain of smart environments. Three different application areas are selected and for each benchmark three different sensor technologies. In order to estimate how popular a certain technology is for a given application, the ACM Digital Library is used to query scientific publications with respective author keywords. This method is limited, as the chosen keywords may not catch all relevant publications. Therefore, the focus is slightly increased by using multiple associated search terms for each application and technology. Additionally, respective searches are also performed using the Google Scholar database that has a much broader scope. The advantages of the latter are the huge collection of scientific resources and no strong selection bias. However, there are various associated issues that may affect the method. The search results vary on the search term, additionally there will be results that mention the search term but do not necessarily rely on the technology for their respective system. Therefore, the results should be considered as an indicator for popularity in the research community. Similar to the ACM DL search different synonyms are considered and an average between the search results is calculated.

The chosen applications are hand gesture interaction, a marker-based identification system and obstacle avoidance for an autonomous system. The technologies are camera systems, radio-based systems, depth or stereo cameras and ultrasound devices.

3.4.1. Scoring

At first the weights of the different applications with regards to the features are determined. The results are shown in Table 3.2. For the tables in this section the short notation of the features is used, in order of their appearance in Section 3.3.1.

Table 3.2.: The importance weighting of different applications, based on the features.

	res	upd	det	unob	proc	robu	disfr	calco	unigd
Hand Gesture	++	++	-	+	o	o	+	+	-
Identification	-	-	++	++	o	++	+	-	+
Obstacle Avoidance	-	+	-	o	+	+	++	++	+

The rating of the different technologies and the resulting score is shown in Table 3.3.

Here it is possible to follow different strategies regarding the rating. In terms of unbiased comparison, looking at the equations it would be necessary that all technologies have the same average feature rating. The second strategy is to apply an absolute ranking to all technologies, independent of the given application. This might lead to certain technologies being unsuited for a given task, or technologies that have the best benchmark score regardless of application. In this specific case the average rating is 0.53 for cameras, 0.58 for radio, 0.44 for depth cameras and 0.56 for ultrasound devices. Table 3.4 displays the different calculated benchmark scores for the

Table 3.3.: Feature rating of the different sensor technologies

	res	upd	det	unob	proc	robu	disfr	calco	uniqd
Camera	++	o	+	-	o	o	o	-	o
Radio	-	+	++	+	o	o	o	o	-
Depth camera	+	o	o	-	-	o	-	o	o
Ultrasound	-	+	o	o	+	o	+	o	o

combinations between applications and technologies. As numerous technologies and applications are compared, the feature-normalized benchmark score is also included.

Table 3.4.: Regular and normalized benchmark score matrix of different applications and technologies

		Camera	Radio	Depth Camera	Ultrasound
Hand Gesture	b_{s_l}	0.53	0.57	0.46	0.55
	$b_{s_l,norm}$	0.50	0.48	0.51	0.50
Identification	b_{s_l}	0.49	0.64	0.40	0.57
	$b_{s_l,norm}$	0.46	0.55	0.45	0.51
Obstacle Avoidance	b_{s_l}	0.47	0.56	0.42	0.59
	$b_{s_l,norm}$	0.44	0.48	0.47	0.53

The effect of the normalization is easily visible. Particularly radio has a high feature rating and is negatively affected by the normalization. The only example with a negative average feature rating is the depth camera. After applying the normalization it becomes competitive in some applications.

Finally, Table 3.5 shows the search results regarding the different technologies and applications. Particularly, the ACM DL keyword search can generate empty results if the search terms are too specific. Thus, the search terms used are “gesture”, “identification” and “obstacle” in this regard and add synonyms for the different technologies. For each sensor category the following synonyms are allowed. “Camera” and “video” for the first technology, “radio”, “rf” and “wifi” for the second, “depth camera”, “stereo camera” and “Kinect” for the third and “ultrasound” as well as “ultrasonic” for the last one. All search results were averaged according to the number of synonyms used. For the Google Scholar search more specific terms were used: “hand gesture”, “user identification” and “obstacle avoidance” with the same synonyms to prevent an excessive number of search results and prevented inclusion of patents and citations. All searches were performed on January 30th, 2014.

Table 3.5.: Search result frequency given specific applications, sensor technologies and synonyms for ACM Digital Library (DL) and Google Scholar (GS)

	Camera		Radio		Depth Camera		Ultrasound	
	DL	GS	DL	GS	DL	GS	DL	GS
Hand Gesture	66	14100	27	7350	32	6850	3	1660
Identification	81	5590	162	4920	10	3957	5	599
Obstacle Avoidance	8	24000	1	13017	17	12278	8	14500

3.5. Results

In this evaluation both benchmark score types are included to outline their differences. “Camera”, “radio” and “ultrasound” have a feature rating above average, whereas “depth cameras” had a lower than average rating. The feature-normalized benchmark score is thus adapted accordingly. Regarding the application of “hand-gesture recognition” this leads to “depth cameras” being considered the optimal technology as opposed to “cameras” that had a higher score before normalization. For the other applications there is no change in optimal technology. The preferred strategy for applying feature-normalized or non-feature-normalized benchmark scoring should depend on the specific benchmarking process. If numerous technologies and applications are benchmarked in a single process, the feature-normalization might be helpful to get a tendency regarding the optimal system. However, if the application is very specific it might be preferred to get a clear ranking and penalize unsuited technologies, regardless of their average feature weight. Accordingly, it is possible to refrain from normalization.

Looking at the search results several conclusions can be drawn. The prevalence is unequally distributed between the different technologies. Both in keywords and general occurrence cameras are the most commonly occurring sensor device, with radio and depth camera ranked behind. Ultrasound on the other hand is less frequently occurring. This may be explained by the higher versatility of the other options. Regarding the “hand gesture” application, cameras have both the highest benchmark scores and most results in the database searches. The benchmark score for “user identification” and “radio” are matched for the ACM DL. However, there are more GS results for “camera”. As already mentioned cameras are more commonly used, yet, the difference in keyword search results is significant. “Obstacle avoidance” is least common in the ACM DL, however quite popular in GS. Accordingly, “ultrasound” sensors are significantly more common in both searches, as opposed to the previous applications. Nonetheless, “stereo cameras” are the most common sensor device for this application. They are commonly used in automotive scenarios, where the detection range of ultrasound is insufficient, as the objects are moving fast [BB98]. Therefore, the application scenario might have to be redefined for fast-moving object detection in open areas as opposed to obstacle avoidance for robots in home scenarios.

Additionally, the method of using database searches for verifying the benchmarking method has to be discussed, as opposed to expert opinion. Surveys of a specific application or certain technologies are common in scientific literature. However, while they might be comprehensive and cite several hundred different applications, the ACM DL database covers more than 2.2 million entries and GS searches can lead to more than 9.7 million results. Therefore, the index searches are preferable in terms of broadness. The search for keywords in ACM DL results in few hits compared to the database size. As they are chosen by the authors there is a large variety in word choice, spelling or number of keywords. While extending the number of different searches might lead to more results overall, it may also lead to additional overshoot, including work that do not cover the desired topics. The GS searches are very prone to overshooting, and should be preferably used to discover trends in data, as opposed to narrowly clustered results.

3.5.1. Central tendency bias

I want to briefly discuss the tendency of the benchmark scores to crowd around 0.5. While the benchmark may result in any number between 0 and 1 the two normalization processes the average is close to 0.5. Thus, even smaller differences close to this average may have a higher significance. This effect is called central tendency bias and is a common occurrence on Likert-scale questionnaires and rating systems [CHE00]. Experts scoring technologies, just like survey respondents have a tendency to avoid extreme responses to a question.

While experience of the person executing the benchmarking process might avoid this problem, it is also possible to use a corrective term in the calculation of the final benchmarking score. The primary purpose of this

corrective term is to make the comparison between different scores easier to the reader. The following equations can be used to fix either regular or normalized benchmarking scores, resulting in the modified benchmarking score m_b , respectively $m_{b,norm}$.

$$m_b = (b_{s_l} + 0.5)^a \quad m_{b,norm} = (b_{s_l,norm} + 0.5)^a \quad (3.10)$$

The exponent a should be a value higher 1 and chosen according to the level of adjustment that is desired. As an example, Table shows adaptations of b_{s_l} and $b_{s_l,norm}$ for cameras and ultrasound, taken from Table 3.4. The different values for a are 1, 5 and 10.

Table 3.6.: Central tendency bias correction for different exponents a

		Camera			Ultrasound		
		a=1	a=5	a=10	a=1	a=5	a=10
Hand Gesture	m_b	1.03	1.16	1.34	1.04	1.28	1.63
	$m_{b,norm}$	1.00	1.00	1.00	1.00	1.00	1.00
Identification	m_b	0.99	0.95	0.90	1.07	1.40	1.97
	$m_{b,norm}$	0.96	0.82	0.66	1.01	1.05	1.10
Obstacle Avoidance	m_b	0.97	0.86	0.74	1.09	1.54	2.37
	$m_{b,norm}$	0.94	0.73	0.54	1.03	1.16	1.34

3.6. Applications for capacitive proximity sensors

In the previous sections I have presented a generic method to evaluate different sensor technologies, with respect to various applications and provided evidence based on analyzing the frequency of technology popularity in scientific literature. In the following section I will use this method to verify potential application domains for capacitive proximity sensors. In Section 2.4 - *Applications in smart environments*, seven different application domains within smart environments have been introduced. Using the specified benchmarking method I will specify the different feature weights and calculate the resulting scores.

3.6.1. Benchmark weights

An overview of the different importance weights associated to the applications is given in Table 3.7. The weights will be subsequently explained in more detail.

Indoor localization does not typically require a high resolution, as it is in most cases sufficient to identify the approximate position of the different actors in the environment. However, a high update rate is required as it is necessary to follow the trajectories of the different persons. Achieving a high detection distance is often the most important aspect, e.g. if it is intended to cover large areas. The system should be unobtrusive and not disturb the actors, while processing complexity is less relevant, since it can be integrated and hidden in the existing environment. The system should be robust. The frequency of disturbances is less of an issue, as indoor surroundings are more controlled, as opposed to outdoor environments. However, if a disturbance occurs it should be calibrated quickly to resume normal operation. The unique disturbances are usually considered in system design and should not affect the overall situation too much.

Table 3.7.: Importance weights of applications for capacitive proximity sensors

	res	upd	det	unob	proc	robu	disfr	calco	uniqud
Indoor localization	o	+	++	+	-	+	o	+	-
Gestural hand interaction	++	++	-	+	o	o	+	+	-
Physiological sensing	++	o	-	+	-	+	-	o	o
Wearable activity recognition	o	-	-	+	+	+	+	o	o
Person-sensing smart appliances	-	+	+	++	o	+	+	+	+
Mobile augmented reality	o	++	+	-	o	+	-	o	+
Obstacle detection	-	+	-	o	+	+	++	++	+

Gestural hand interaction requires high resolution and high update rate, as it is intended to capture fine movements of the fingers. The detection distance is not critical, since it is assumed that the capturing device can be put in close proximity of the hands. Most systems are designed for explicit interaction and it is not necessary to be unobtrusive. Likewise, enough processing capabilities are typically included. The system should be fairly robust, as reliable input is expected and it should not be disturbed, resulting in unwanted actions. The calibration in case of disturbance should be ensured, or at least a feedback has to be given to the user. There are no critical unique disturbances in this specific application.

Physiological sensing often has to work on weak signals produced by the body and thus needs a high resolution. The required update rate likewise should be high as several of those signals are fast, such as heart rate or brain functions. The detection distance is weighted least favorably, as they mostly require a direct touch. However, even then it should be fairly unobtrusive, leading to wearable systems that have restricted processing capabilities, but nonetheless should be very robust. The signals are typically noisy with high frequency of disturbance, yet recalibration is not very costly and due to the touch properties no particular unique disturbers should be present.

Wearable activity recognition is mostly performed by wearable fitness trackers that have to be very conscious about energy usage. Thus the resolution is average and the update rate low. The system has to be close to follow the movements directly. However, it is preferable if there is an integration into other devices, or a small form factor, so they are less obtrusive. Processing capabilities are limited, yet the design should be very robust to avoid damage during typical activities. Disturbance mostly occurs due to unforeseen movements that can be misinterpreted, though are not very difficult to calibrate. Likewise, there are no particular unique disturbances apart from movement that we have to consider.

Person-sensing smart appliances are designed to just detect the presence of a human being, without providing any additional information apart from its own location. Therefore, the requirements for resolution are low, while this information should be available with a high update rate and preferably over a fairly large detection distance. The systems should be unobtrusive and robust, while the processing complexity of the signal is not critical, given the simple measurements that can be handled even by very low-powered devices. The systems should distinguish between persons and other moving objects, thus being reliable towards unique disturbances that may occur often. A recalibration can be fully automated.

Mobile augmented reality is typically executed on a smart phone. The localization is based on GPS using the limited resolution. However, the other sensors should clearly and precisely identify orientation, leading to mixed requirements on resolution. The update rate and detection distance has to be high, particularly regarding the orientation estimate. The systems can't be unobtrusive, as an additional device is needed to overlay the visual information. The processing complexity is average and can be performed even by slower smart phones. The

system should be robust and provide its service at all times. The system is difficult to disturb, can be easily calibrated and there are no obvious unique disturbers that have to be considered.

Obstacle detection is a primary concern of autonomous systems operating in a smart environment. The resolution is less critical than the update rate, as there are various strategies to deal with encountered obstacles, but they have to be detected swiftly and robustly. Thus a higher detection distance is preferred. It is not necessary to be unobtrusive and it can be assumed that enough processing capabilities are present. For this application we have to account for disturbing factors, as it is mandatory that the system detects obstacles of all materials, shapes and most sizes. For most sensor categories there are materials that will prevent a proper detection. Thus unique disturbers have to be factored in, according to their prevalence in the given scenario.

Capacitive proximity sensor feature rating

As the next step in the benchmarking process, the feature weights of capacitive proximity sensors have to be determined. Following the same process used for the different technologies discussed previously the result can be seen in Table 3.8.

Table 3.8.: Feature weights for capacitive proximity sensors

	res	upd	det	unob	proc	robu	disfr	calco	uniqd
Capacitive proximity sensors	-	+	-	++	<i>o</i>	<i>o</i>	<i>o</i>	+	-
Camera	++	o	+	-	o	o	o	-	o
Radio	-	+	++	+	o	o	o	o	-
Depth camera	+	o	o	-	-	o	-	o	o
Ultrasound	-	+	o	o	+	o	+	o	o

Looking at the group of sensor performance characteristics the resolution has to be considered coarse. While it is possible to create highly sensitive and precise sensors, this is typically restrained to very close distances. The majority of capacitive proximity sensor applications are operating over a certain distance. However, the sensor layouts that allow detection in this area do not have the resolution to precisely distinguish precise object locations without extensive post-processing. There is no set limitation to the update rate of capacitive sensors, as also high-frequency electric circuits can be evaluated, e.g. we created an exemplary application that has an update rate of about 1kHz [GPBB*13]. This will affect the achievable resolution and distance. Most setups operate in the range between real-time 20Hz up to 100Hz leading to a favorable update rate. The detection distance of the sensors is limited. While there are some systems that operate in distances of more than 1m [Mac04], they are restricted in their sensing abilities. Typically the systems are operating in a range somewhere between 10cm and 50cm.

Discussing briefly the pervasive metrics, the major advantage of capacitive proximity sensors is their ability to be installed completely invisible. They also operate in a frequency spectrum that is not considered biologically active and at low voltages. The processing complexity varies, based on applications. They will require pre-processing and calibration that is more complex compared to other sensors. However, as there are not many sensor value the post-processing is simpler, e.g. compared to cameras. Some data processing methods can use fairly complex statistical methods if multiple objects are to be tracked or there is additional gesture recognition [GPBKK13]. Overall, the processing complexity is rated average. Similar to the other sensor systems presented robustness and quality of service depend on the particular application. It is possible to design systems that fail easily, or that are fully redundant. In consequence, they receive an average rating.

Finally, the environmental characteristics are discussed. The frequency of unique disturbances is average. The sensors can be disturbed by electrical signals that are present in the environment. The majority of potential disturbing signals can be compensated in an initial calibration phase and by using appropriate measures to isolate the measuring system from the existing electric circuits. The remaining disturbances are either caused by devices that are brought into the environment after, or by irregular disturbances in the electric supply. E.g. in our laboratory we had a persistent issue caused by a faulty power supply in a neighboring building that was affecting the stability of the supply frequency. The complexity of calibrating the sensors is low. There is no need to use any external measure and it is sufficient to consider a time-series of previous measurements for calibration. The example above about a single power supply disturbing all capacitive sensors is one example why unique disturbers have to be taken into account. While these instances are limited the rating is reduced to "not favorable".

3.6.2. Benchmarking scores for capacitive proximity sensors

Using the methods of the previous pages it is now possible to calculate the different benchmark score for capacitive proximity sensors and the various applications. In this case there is no comparison between different technologies. To provide a complete overview all four varieties are calculated and discussed. The results are shown in Table 3.9.

Table 3.9.: Benchmarking scores for capacitive proximity sensors in different applications

	b_{s_l}	$b_{s_l,norm}$	m_b (a=10)	$m_{b,norm}$ (a=10)
Indoor localization	0.56	0.53	1.74	1.31
Gestural hand interaction	0.58	0.55	2.08	1.56
Physiological sensing	0.57	0.54	1.89	1.42
Wearable activity recognition	0.55	0.52	1.63	1.23
Person-sensing smart appliances	0.57	0.54	1.97	1.48
Mobile augmented reality	0.50	0.47	1.00	0.77
Obstacle detection	0.55	0.53	1.70	1.28

As seven different applications are analyzed without comparison of different sensing technologies, the normalized scores have the purpose of putting the calculated scores around the 0.5 average feature rating. Additionally, the correction term for central-tendency bias is included for both regular and normalized scores, with a weighting factor a of 10.

The lowest score is associated to mobile augmented reality, with a normalized unbiased value of 1.00. This application requires localization over long distances and orientation detection of the device. Capacitive proximity sensors are not able to add anything to GPS and magnetometers here and thus have not been used in literature for this specific task. Another low score has been designed to obstacle detection, with unbiased score of 1.28. Capacitive proximity sensors are not able to detect ungrounded objects, which is critical in many instances of obstacle detection. Mobile robots will need to detect stone walls, which is not possible by capacitive sensors alone. However, they could augment other obstacle detection systems in detecting conductive objects.

Indoor localization has a score similar to obstacle detection, with an unbiased score of 1.31. This application is known for capacitive systems in literature. The scoring here is favoring systems that achieve a high detection distance, based on the majority of applications being based on sensors that are attached to the walls or based on wearable devices the user is carrying around. Capacitive localization systems are typically installed in the floor, e.g. TileTrack [VMV09] and SensFloor [LS09]. Therefore, they don't need a high detection distance as persons

will be walking above the floor. In this case the presented model would have to be adapted, e.g. by distinguish floor-based localization systems and tag-based localization systems. The disadvantage would be a less generic application domain.

There are three further applications that have a similar score. Gestural hand interaction has an unbiased score of 1.56 and is one of the most common applications of capacitive proximity sensors in literature, having been used in a variety of different systems. Physiological sensing has an unbiased score of 1.42. The limiting factor here is the high requirement on resolution. This can be overcome using large sensors, or sensors that are very close to the human body [CAL10]. Wearable activity recognition has a high unbiased score of 1.63. This is an example of a topic that has not been researched extensively. The Active Capacitive Sensing concept by Chang et al. just presented for physiological sensing has also been applied to activity recognition [CAL10]. Another example is a concept to augment accelerometer-based activity recognition with capacitive proximity sensors, presented by Grosse-Puppenthal et al. [GPBB12]. Both of the publications are fairly new. Thus this is an application area that seems suitable for capacitive proximity sensors that could lead to further research.

Finally, the highest unbiased score is achieved for person-sensing smart appliances with 1.97. This application area is particularly well-suited for capacitive proximity sensors. The feature weights and importance weights match well. There are also numerous examples in literature that realize this scenario, such as the NEC passenger seat [Smi99] or HandSense by Wimmer et al. [WB09].

Based on this scoring and the note examples from literature I have investigated four different application domains for capacitive proximity sensing:

- Indoor localization
- (Person-sensing) smart appliances
- Physiological sensing
- Gesture Interaction

3.7. Discussion

In this chapter I have introduced the benchmarking model that calculates a benchmark score that is an indicator for the suitability of a sensor technology for a certain application. Additionally, it is possible to use the inverse option and benchmark a single sensor technology for a number of applications. The model was derived based on a set of common features for sensor technologies and a weighting factor determining their importance for smart environment systems. I have applied the method to capacitive proximity sensors and the different smart environment applications introduced in the related work. This allowed me to identify the application domains that are most suitable for capacitive proximity sensors and discard others that have a lower benchmark score. In the next chapter this collection of application domains will be used to identify a set of distinct use cases for capacitive proximity sensors systems. Each of those has unique challenges that will be identified. They can be approached using a variety of processing methods that are subsequently discussed.

4. Use cases for capacitive proximity sensors

The last chapter outlined the potential application domains for capacitive proximity sensors. In the following Following I will derive actual use cases for the technology. For each of those I present associated challenges and new data processing methods that are specifically targeting these. Next, I present a number of prototypes that are implementing these methods. This allows me to gather sufficient knowledge to the application of capacitive proximity sensors in smart environments and validate their use in Chapter 5. In addition to the use case specific challenges, there are also numerous considerations regarding the implementation. These are detailed in the descriptions of the specific prototypes.

4.1. Use cases and associated challenges

Looking at the previously defined application domains for capacitive proximity sensing, we can get more specific and derive actual use cases that belong in the different domains. While it would be possible to associate the different systems presented in the related works, this approach would not allow to capture actual challenges in the application process, apart from the discussion within the specific work. Thus, I am focusing on the implemented prototype systems that will be presented in the subsequent sections. Table 4.1 shows the different application domains, how capacitive proximity sensors can be applied, and the use cases that were derived from it. In this section I will discuss how this table was derived and create a list of challenges that become apparent when designing the specific systems. Based on these challenges it is possible to identify a number of improved and new methods in the processing of capacitive proximity sensor data that are required to design the presented applications. These specific contributions to the processing methods will be detailed in the following section.

Table 4.1.: Application domains and derived implemented use cases for capacitive proximity sensing

Application Domain	Using capacitive proximity sensors	Implemented use cases
Indoor Localization	Sensing system hidden in environment	Capacitive sensing below floor cover
Smart Appliances	System detecting presence and other parameters of human bodies in range	Posture recognizing office chair, occupation sensing bed, arm detecting arm-rest
Physiological Sensing	Determine physiological parameters associated to movement	Breathing rate detection via chest movement, long-term movement analysis
Gesture Interaction	Hand interaction in near range	Finger gestures, single hand 3D gestures, combined multi-hand and touch tracker

Indoor localization has been presented as one of the application examples for smart environments in the related works section. The main advantage of capacitive systems is their unobtrusive application in the environment as presented in TileTrack [VMV09] and SensFloor [LS09]. Capacitive indoor localization systems can be hidden below any non-conductive material and enable tracking of users on a distance. Particularly interesting is the ability to place the sensing equipment below the floor cover as an additional layer, e.g. when installing a new carpet or wood parquet. SensFloor as commercially available solution is designed to be placed as an additional layer and integrates sensors and communication systems therein. While this enables precise sensing close to the walking persons, it is costly and can lead to maintenance issues once the sensors integrated in the layer fail. Instead, it is also a viable option to separate sensor hardware and electrodes, e.g. by placing the sensors on the borders of the indoor area and use a specific electrode layout below the floor. The electrodes can be made of any conductive material and can be insulated using non-conductive isolation to prevent corrosion and physical damage. The system components that are most prone for failure are the connections between electrode and sensor and the sensor hardware and its communication channels. Those can e.g. be placed within the border covers.

The main challenge of this solution is to balance the number of sensors and the intended resolution. Using a limited set of sensors placed on the border it should nonetheless be possible to determine the positions of one or more persons on the area above. Preferably additional information should be gathered, e.g. if a person is standing, sitting or lying. In addition to the cost factor, a larger number of sensors also causes several other issues. One example is cross-talk between the different electrodes that has to be avoided using a variety of multiplexing methods. The achievable resolution of the single electrodes is depending on several factors, including the measurement time, the applied voltage, distance between electrodes and floor surface, or the geometric layout of the electrodes. Thus, it is important to find an electrode layout and processing methods that achieve this balance.

Smart appliances as presented in the related works comprise a very diverse group of devices that are in the current environment. There is a huge variety of sensor categories and processing that can be applied to any given task. Looking at capacitive proximity sensors, the major advantage is the invisible application that allows to create smart appliances that are indistinguishable from non-equipped systems. Using different conductive materials for the electrodes this integration can range from solid antennas hidden within the appliance to conductive threads that can be woven into fabric. The main application for capacitive proximity sensors in smart appliances is the sensing of different parameters of persons interacting with the system. For example the sensors can be used to recognize the posture of a user and use it to adapt certain parameters of the appliance or the environment. This type of interaction has also been called implicit interaction, as the user does not directly attempt to manipulate the environment, but instead the activities are interpreted as input according to the given situation [SV01]. In many instances it is sufficient to get information about the presence of the user. A simplified posture recognition can be used to detect presence or occupation, based on the data acquired by one or more sensors. Finally, it is often also interesting to detect if certain body parts are currently at a given location, e.g. the arm resting on the armrest of a chair, indicating a specific situation that the system can react on.

The challenges in this domain are manifold. Existing posture recognition systems might rely on a different sensor category, supporting hundreds of measurement spread over a larger area. Again, capacitive proximity sensors are distributed sparsely and need methods that enable gaining a similar amount of higher-level information. Here it is necessary to create models of the human body that are suited for the processing of capacitive data. According to the parameters that are supposed to be detected the models can be more or less complex and thus optimize the required processing time. A sensing bed that has to detect if a person is lying would require a simpler model, as opposed to a sensing chair that has to detect a larger variety of different postures. Often the capacitive systems are combined with other systems, or use a custom non-uniform distribution of electrodes in the device that require methods of data fusion and processing of heterogeneous signals to acquire higher level information, e.g. an armrest that combines the detection of an arm and an interactive area that allows gesture interaction.

Physiological sensing allows us to measure signals generated by the different process of the human body. One common application the tracking of training effects by athletes. This may include measuring heart rate or respiration. There are numerous medical applications ranging from long-term blood pressure sensing, to blood glucose level sensing or tracking the quality of sleep throughout the night. Additionally, there are physiological signals derived from long-term monitoring, such as movement-based sleep-phase detection. Measuring electric properties is the most common variety to detect physiological parameters. This includes EEG, measuring the brain activity, ECG, measuring the heart rate, or sensors for skin conductance that can infer the stress level. For all these applications electrodes are placed very close to the measured property, often even requiring contact. Capacitive proximity sensors on the other hand are used over a distance. The systems can be designed to enable a high resolution that can track very small movements of the body. Rob MacLachlan has created a spread spectrum system that is able to measure the chest movement associated to breathing over a distance of more than 30cm [Mac04]. Other examples include the detection of swallow movements [CAL10].

There are various challenges when trying to gather physiological signals from capacitive proximity sensors. A major problem is to distinguish the measured property from other signals generated by movement of the body. Here it is possible to use the effect that the measured properties often are prevalent in a specific frequency range. Thus, if the signals are analyzed in the frequency domain it is possible to extract the physiological properties from the overall signal, e.g. when analyzing the chest movement associated to respiratory rate and focusing on the most important frequency intervals. Regarding long-term physiological signals, capacitive proximity sensors can be used to aggregate data on movements. In this regard, it is interesting how the sensor data in time-domain can be associated to particular movements that can be used in long-term analysis of the user's physiological patterns, e.g. to detect sleep phases. Based on the particular setup of the system a different set of features has to be selected and evaluated.

Gesture interaction is a very diverse application area that reaches from the acquisition and interpretation of whole body gestures to small movements of the fingers registered on surfaces. It is maybe the most thoroughly researched domain for capacitive proximity sensors. It started with Leon Theremin's musical instrument that was controlled by fine movements of the hand. The MIT research group experimenting with capacitive interaction in the 90s created some concepts for touchless interaction, e.g. the Field Mouse that allowed to track position and orientation of two hands to enable 3D interaction [Smi96], or an art installation that could be controlled using a set of gestures [SWD*98]. A new category of interaction devices such as Wii remote, Kinect or Leap motion led to the proclamation of more natural interaction between human and machine [Val08]. While capacitive touch sensors have become ubiquitous in mobile devices, the proximity variety is less frequently used. Wimmer integrated several sensors into a table to enable a regional interaction on the surface [WHKS06].

While the area has been well-researched there are still several challenging aspects. In many instances the capacitive interaction devices will have a different resolution depending on the direction of the approaching object. In the last years there has been a rise in methods that allow a generic recognition of gestures in two dimensions, e.g. from mouse cursor movement and finger movement on touch screens. It is interesting to evaluate if this is also possible for 3D positions acquired from capacitive proximity sensors. The acquisition of the hand position is also challenging, as the sensors can't distinguish between hands and other conductive parts of the body. Thus it is necessary to investigate different methods of fitting arms and hands, particularly on larger area interaction devices. Additionally, it is challenging to enable gestures via multiple hands and arms. In many gesture interaction applications fatigue may occur if the hands have to be moved too much, or the arms have to be held in free air for a longer period. Thus it is necessary to design specific graphical user interfaces that are suited for this form of interaction. If the capacitive proximity sensors are placed under thicker layers of non-conductive material it is difficult to detect touch events from capacitance data alone. It becomes interesting to combine capacitive proximity sensors with other sensor categories that can detect touch or even different touch events, thus allowing a richer interaction.

Table 4.2.: Challenges associated to the different use cases for capacitive proximity sensors

Use cases	Challenges
Capacitive floor sensors	Sparse sensor distribution in large areas, geometric electrode layout
Posture chair	Multi-body models, electrode material
Occupation sensing bed	Single-body models, movement tracking
Armrest supporting gestures	Heterogeneous capacitive arrays
Breathing rate detection	Frequency spectrum analysis
Sleep phase detection	Long-term movement features
Finger micro gestures	Small 3D movements
Multi-arm tracking	Arm and hand fitting, interaction design
Combined touch sensing	Combining position tracking and touch events

In consequence, there is a large number of specific challenges that can be tackled in the different domains. They can be associated to the identified use cases using Table 4.2. In the following section I will present the methods that contribute to the different challenges in processing the data generated from capacitive proximity sensors.

4.2. Processing methods

After analyzing the challenges associated to the different use cases for capacitive proximity sensor I will use the following section to specify a number of novel or improved data processing methods that can be used in this context. They are grouped into five specific areas.

Sparingly distributed sensor arrays refer to configurations that have a limited number of sensors spread over larger areas. In this regard, it is important to find methods that allow acquiring sufficient information about the object to be detected. Typically information about the electrode geometry and interpolation methods are used to meet these requirements.

The second area is model-driven fitting methods. Using a simplified model of the object to be detected, it is possible to fit these to the received sensor data. The area can be distinguished according to the complexity of the models that are either comprised of a single body or multiple parts that are connected to each other.

Heterogeneous sensor systems can be described as a combination of multiple sensors that are not uniform. In terms of capacitive proximity sensors this can refer to either arrays of different capacitive sensors that use a geometric layout of varying sizes and shapes or the combination of capacitive sensors with other categories of sensing systems in a meaningful fashion.

Image-based processing describes the method of creating an image from capacitive sensor data and applying different algorithms associated to visual computing. An uniform array of capacitive proximity sensors resembles an array of light sensors, albeit acting on a different frequency interval of electromagnetic radiation. Thus, with a few limitations it can be treated similar to a camera system with operations applied on a pixel level.

The last group is the processing of physiological signals in time domain and frequency domain. Many physiological activities rely on the movement of muscles, e.g. the beat of the heart or the chest movement associated to breathing. These movements have an effect on the electric field generated by capacitive proximity sensors and can be analyzed using a variety of different methods.

4.2.1. Sparsely distributed sensor arrays

Sparsely distributed sensor arrays are layouts that limit the number of available sensors, either by environmental parameters or by design. This limits the information that can be gathered about the detectable object, or reduces the number of different objects that can be distinguished. To compensate this limitation a number of interpolation methods can be used that take into account our knowledge about the position and shape of the electrodes that are used in the current setup. One example for this sparse distribution is the previously presented Thracker system that uses only four electrodes to acquire a hand position and detect gestures at certain positions [WKBS07]. In this section I will present two different contributions - a new method to recognize single-hand gestures in free-air, using just six different sensors, and an indoor localization system based on a coarse grid of wire electrodes that can be hidden below different floor surfaces.

4.2.1.1. 3D location tracking and gesture interaction

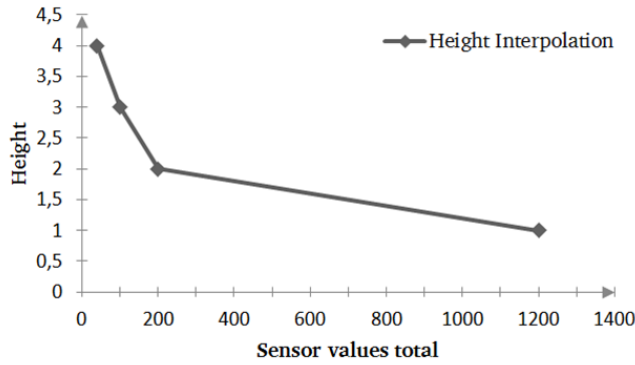


Figure 4.1.: Piecewise linear hand distance estimation [BH11]

Gesture recognition can comprise a large number of different body movements, including sign language that uses movements and position of hands and fingers, the posture of the whole body, or as in our case the movement of the hand in three dimensions. This requires two distinct processing steps. At first it is necessary to precisely localize the position of the hand in the interaction space. Afterwards, a time series of these positions has to be analyzed and attributed to different gestures. The localization method was first presented in a publication from 2011 [BH11]. The static gesture recognition method used there was later extended by adapting algorithms used to detect mouse gestures for movements in three dimensions [BDK13]. The first data processing step is the planar localization of the hand, following a weighted average algorithm, whereas n is the number of sensors, (x_i, y_i) the location of the electrode centers and v_i the value of the given sensor.

$$\bar{x} = \frac{\sum_{i=1}^n v_i \cdot x_i}{\sum_{i=1}^n v_i} \quad \bar{y} = \frac{\sum_{i=1}^n v_i \cdot y_i}{\sum_{i=1}^n v_i} \quad (4.1)$$

In order to calculate the distance of the hand from the plane, piecewise linear interpolation is used that resembles the response curve of a single sensor [BH11]. In this case four different thresholds t_i are used to calculate the proximity, based on the sum of sensor values. t_1 indicates the closest distinguishable proximity, e.g. touch, with all higher value sums associated to this. t_4 represents the maximum distance in which the sensors can detect a hand. One example with fixed points at values 40, 100, 200 and 1200 is shown in Figure 4.1.

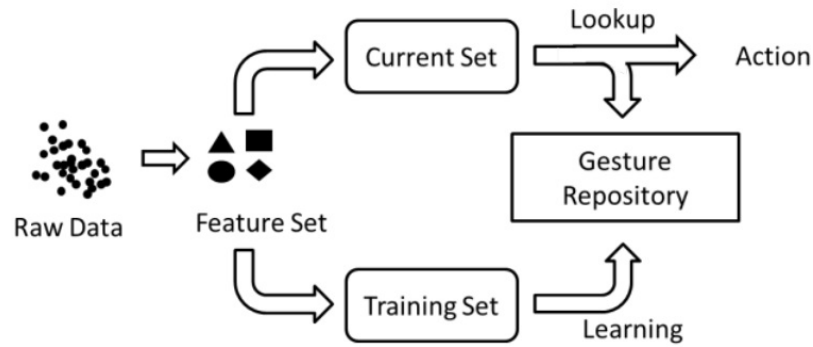


Figure 4.2.: Principle components of a learning by example recognition framework [BDK13]

Initially a static gesture recognition method was implemented that used a series of five subsequent locations and simple heuristics to determine a small number of gestures. However, this limits the potential gesture set and is difficult to extend. Thus, a generic gesture recognition module was created, based on learning by example [BDK13]. The general functionality of a gesture recognition framework based on learning by example is shown in Figure 4.2. A feature set is extracted from incoming raw data. Collections of these are distinguished into training sets that are used to associate certain features to given gestures. After a learning process the current feature sets that are acquired on-the-fly, are tested against the training sets in a repository. These look-ups can lead to successful gesture recognition and association to certain actions. This association method is also called classification. There are numerous approaches, e.g. neural networks (NN) or support vector machines (SVM).

4.2.1.2. Large-area location tracking

There are several systems that use capacitive proximity sensing to track the location of one or more persons in an environment. A common challenge in large areas is achieving a suitable coverage with electrodes at all positions. Lauterbach and Steinhage overcome this problem in their SensFloor system by integrating sensors and electrodes in an underlay that can be placed below the upper layer of the floor [LS09]. TileTrack, the system developed by Valtonen et al. requires large emitter electrodes under the floor and receivers placed in the walls [VMV09]. This limitation was reduced in a later iteration where they integrated receivers into different pieces of furniture.

While SensFloor allows to cover large areas by having the sensing electrodes near all surface areas, a limitation is the maintenance. If a sensor breaks below the floor covering it is difficult to replace. TileTrack in its initial installation requires proximity to walls, or later to specific pieces of furniture that are in the environment. This is difficult to guarantee in many instances and requires an initial calibration of the environment, according to placement of the furniture.

I propose a system based on a rectangular grid of long wire electrodes that are placed below the top floor cover. The sensors are attached at the edge of the area, e.g. in skirting boards [BHW12]. As the system is based on loading mode, there is no need for dedicated receivers, but instead it relies solely on the electrodes below the floor. The system is akin to a larger variety of projected capacitive touch screens that are partially also using grid layouts. However, those typically rely on shunt mode measurements [BO10a].

Using long, straight wire electrodes has different effects on the measurement. One effect of this is the limited detection distance that is not comparable to large plate electrodes. Particularly if thick floor covers are used, the grid has to be fairly dense. Another effect is the sensitivity towards noise and influence from outside electric

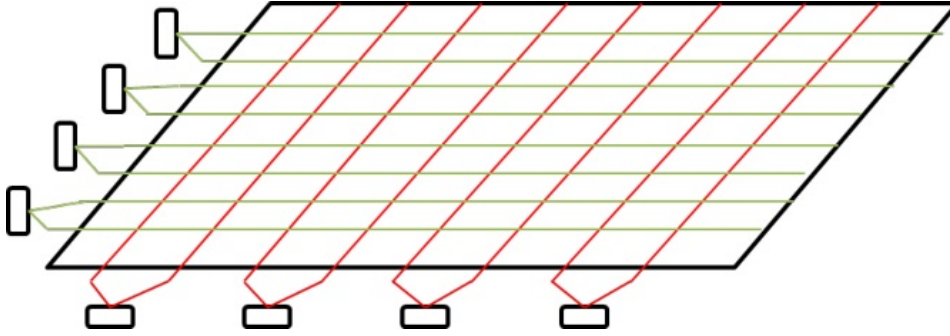


Figure 4.3.: Wire electrode grid below floor cover attached to sensors on the border

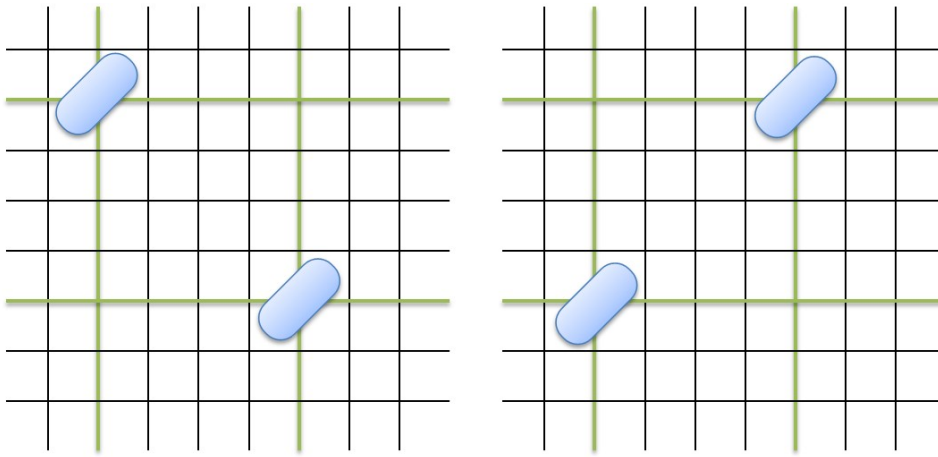


Figure 4.4.: Two potential person locations resulting in same sensor readings (green indicates active electrodes)

fields. Therefore, the system requires preprocessing to reduce the noise and achieve a more robust high-level data processing. In order to localize persons, the system uses an adapted weighted average algorithm, similar to the variety presented for the gesture recognition in the previous section. Each electrode is considered to only have a single coordinate in either x or y direction, allowing to easily calculate the center-of-gravity. However, it can occur that only two electrodes are active at a certain point in time, while two persons are present and too far away from any other electrode to be detected. In these cases there is a certain ambiguity as each (x, y) value combination can result in two potential intersection points, as shown in Figure 4.4. To overcome this problem, it is possible to either use a mix of sending and receiving electrodes operating in shunt mode and specific measurement cycles, or analyze the time-series of previous locations to discard unlikely positions.

Similar to SensFloor, the concept also supports detecting additional information about the persons present, most notably fall detection. This is based on a time-series analysis of aggregated values of the sensors that are currently detecting an object. This method is using the assumption that the overall sensor response is roughly equivalent to the shape of the object that is closest to the surface. This results in a higher capacitance of the overall system, similar to the plate capacitor model. The effect is shown in Figure 4.5. The sum s of all n sensor values r is the closest equivalent to the system capacitance and therefore a viable measure. If the overall value is

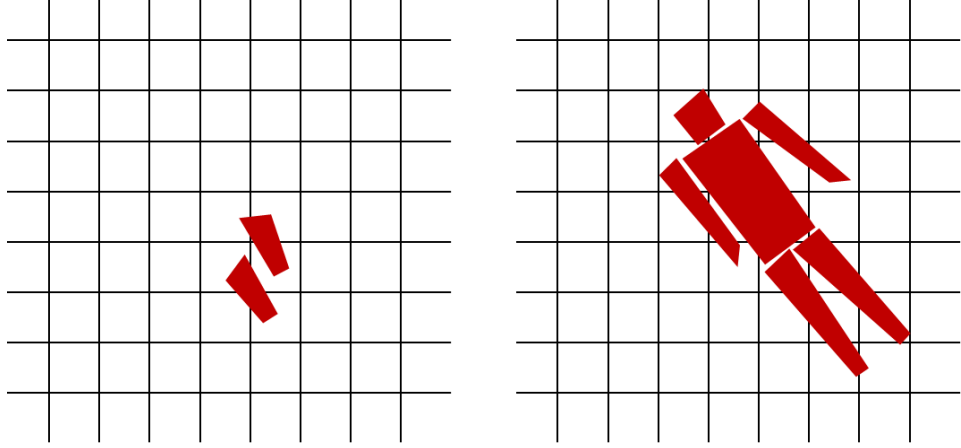


Figure 4.5.: Shapes of a standing and lying person on top of the CapFloor grid

beyond a certain threshold v_l , we can consider a lying person p_l .

$$s = \sum_{i=0}^n r_i \quad p_l = \begin{cases} 1, & s \geq v_l \\ 0, & s < v_l \end{cases} \quad (4.2)$$

In order to increase the robustness this threshold has to be exceeded for a certain amount of time t_m . In consequence a fall f is detected if the following equation is 1.

$$f = \prod_{j=0}^{t_m} p_{l,t_j} \quad (4.3)$$

4.2.2. Model-driven fitting methods

When acquiring sensor data from physical objects, it is often difficult or even impossible to analytically describe the resulting value. There are numerous environmental factors influencing the signal and the properties of the object might not be clearly determined. Considering the human body, there is a mostly unconstrained number of sizes, shapes and biological properties that influence the response to an electric field. Thus, in order to fit sensor outputs to the potential object configurations, simplified models can be used that resemble the actual physical effects and can be described analytically. Regarding capacitance of the human body relative to a single sensor, a common abstraction is a sphere having a diameter close to the height of an average human [Sea97].

Models based on a single geometric objects are considered single-body, while connected geometric objects that comprise a single model can be called multi-body. Smith used a model of multiple spheres to approximate arm position and rotation above an array of capacitive proximity sensors [SWD*98]. Another possibility is adapting the models to a derived physical effect. Harada et al. are using the projected pressure distribution of a virtual skeleton and body model on a flat surface to create a pressure distribution that can be compared to the actual pressure effect generated by an actual human body resting on a set of sensors [HSM00]. In this section I will describe two novel methods to fit abstracted models of the human body to sensor readings acquired from smart furniture systems. The first method uses a cylindrical human body model to match the posture of one or two bodies on a bed, the second method uses a multi-body skeleton that is fitted to sensor readings determining posture on a chair.

4.2.2.1. Single-body models

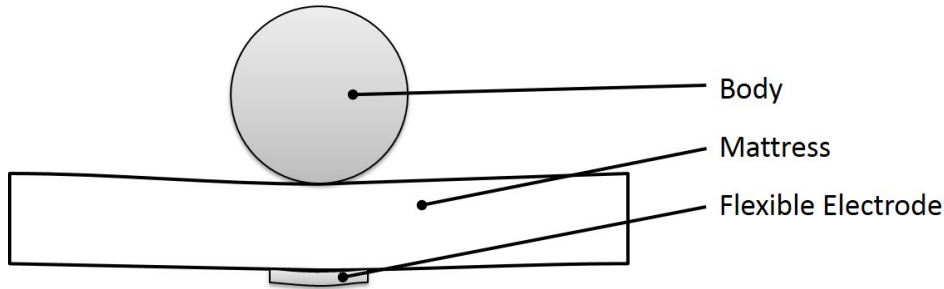


Figure 4.6.: Object on mattress decreases distance and changes geometry of flexible electrode [BH12]

While models are more often applied directly to the capacitive sensor data, it is also possible to use a different property that is derived from it. Together with Henning Heggen, I developed a system that uses flexible electrodes that bend on pressure. In previous experiments we observed that bending a calibrated flexible electrode results in a higher value of the sensor. Thus, the idea was created to use this effect for combining presence and pressure detection. A first application was to use such a system on the slatted frame of a bed. The following section is based on two different publications [HB10, BH12].

If an object is applying force to the mattress there are two cumulative effects. The object will be in detection range of the sensors, decreasing the distance by deforming the mattress and the applied pressure changes the geometry of the flexible electrode, resulting in a higher sensor value. The basic idea is shown in Figure 4.6. If a model is used that approaches the pressure distribution of a human body, a small number of capacitive proximity sensors equipped with flexible electrodes should allow to detect different posture and occupation configurations on the bed. Additionally, sitting and lying can be distinguished, for one or two persons. While the sensor values are increasing according to higher pressure, this relationship is highly non-linear and influenced by three groups of system parameters. The sensor system parameters include the sensor design, e.g. excitation voltage and frequency, as well as geometry and material of the electrode. These parameters additionally determine range and precision of the overall system. The second group are environmental parameters that influence the conductivity of the excited electric field, including humidity and temperature. The third group are the bed parameters. These include the hardness of the slatted frame that influences the geometric deformation of the affixed electrodes and the mattress parameters, comprised of hardness, thickness, material and type, influencing how far the body is away from the electrodes and how the pressure is distributed onto the slatted frame.

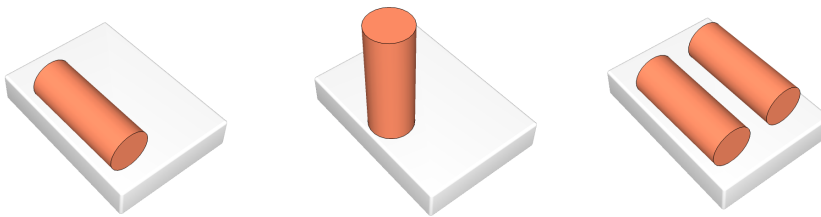


Figure 4.7.: Cylindrical human body model and various poses on mattress [BH12]

To identify occupation and positioning we use a very simple model for estimating the effect of a human body on the sensor values. As previously mentioned, the sensors react to both presence of a body and applied pressure. The human body is modeled as a cylindrical object on the mattress. The object can be either sitting or lying and there might be multiple persons on a single bed. A few potential poses are shown in Figure 4.7. It is assumed that a sitting person will cause a high pressure within a small region, and that a lying person has a moderate pressure distributed over a larger area. Thus, the challenge of determining posture and orientation from a limited number of capacitive sensors, tuned to detect pressure, can be formulated as an inverse problem. If we assume a constant density of the cylinder and a uniformly deforming mattress, the idealized pressure distribution is uniform as shown in Figure 4.8.

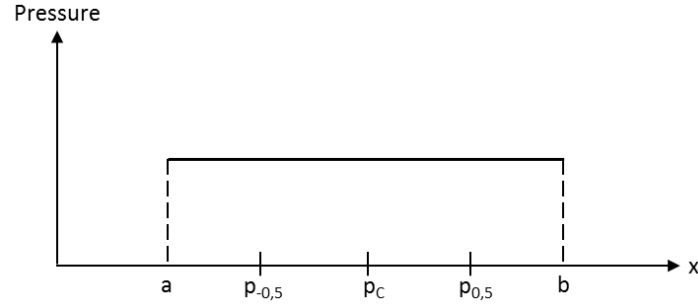


Figure 4.8.: Pressure distribution of a uniform cylinder [BH12]

The system is further simplified by reducing the significant values of the pressure distribution to just two distinct values. Be p_c is the center of pressure and $p_{-0.5}$, $p_{0.5}$ the points enclosing half of the pressure distribution. The center of pressure and standard deviation σ are sufficient to describe the system, as shown in the following equations:

$$p_c = \frac{a+b}{2} \quad \sigma = \sqrt{\frac{(b-a)^2}{12}} \quad (4.4)$$

$$p_{-0.5} = p_c - \sigma \quad p_{0.5} = p_c + \sigma \quad (4.5)$$

The raw data from the sensor is considered as random, uniform sampling, a discretization of the continuous distribution. The center of pressure and the standard deviation are calculated using the geometric information, the position of the sensor \vec{x} .

$$p_c = \frac{\sum_{i=1}^n v_i \vec{x}}{\sum_{i=1}^n v_i} \quad (4.6)$$

$$\sigma = \sqrt{\frac{1}{n} \left(\sum_{i=1}^n \vec{x}^2 - \frac{1}{n} \left(\sum_{i=1}^n \vec{x} \right)^2 \right)} \quad (4.7)$$

Using this model, a set of potential poses that cover the most common situations has been determined. Potential poses for one and two occupants are distinguished. One person may sit at a certain location or lie on the bed

in various angles. It is assumed that the head is always at the upper part of the bed. Two persons may either both sit, both lie down, or one is sitting while the other is lying. The limitations of this model concerning the actual system are the non-uniform pressure propagation throughout the mattress, as well as the non-linear sensor response on different pressure levels. Therefore, it is not expected that the deviations adhere to the theoretical model, but instead configurable thresholds are used that allow for increased robustness in exchange for precision.

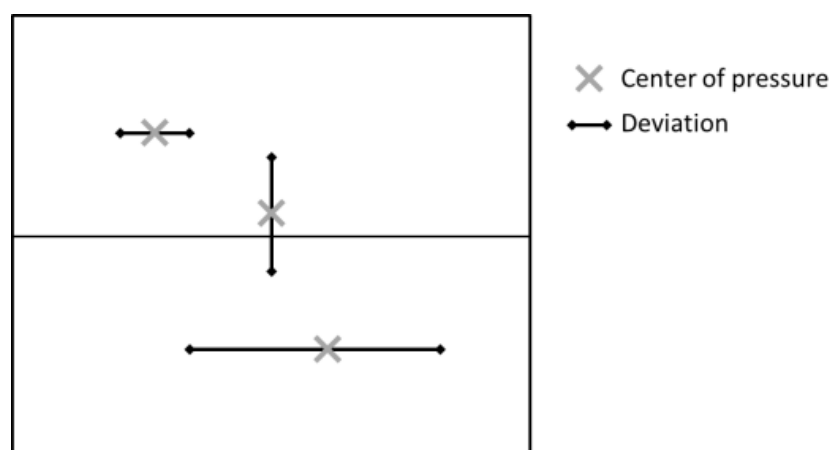


Figure 4.9.: Calculating centers of pressures and deviation [BH12]

Occupation and posture detection are performed by dividing the two person bed into a left and right section. For each side the total sensor values, assumed center of pressure and the standard deviation are calculated individually (Figure 4.9). The same calculation is performed between the two areas, allowing to distinguish the specific side of activity, or detecting persons that are lying on the bed diagonally. Using these six intermediate values it is possible to map the different poses. If all activity is on one side and the horizontal deviation is low, it can be assumed that one person is sitting. Additionally, the intermediate values can be used to gather more information, e.g. the exact location a person is sitting at.

4.2.2.2. Multi-body models

If it is necessary to identify more complex human behaviors than sitting and lying, a single body model is no longer sufficient. In the briefly mentioned work by Harada et al. the underlying model was a 3D representation of an average human body that could move various joints freely [HSM00]. This allows to detect a variety of different postures, in this case using an iterative process based on potential energy, momentum and difference between the actual and simulated pressure distribution. Using such full body models, based on an internal skeleton of connected joints is also common in full body gesture tracking systems. However, while in those cases the volume of the different body parts is important, there is not necessarily any additional physical simulation of properties, such as weight and density [SFC*13].

In a project together with student Sebastian Frank, a concept for a smart chair was developed that unobtrusively integrates a set of capacitive proximity sensors to detect presence, posture, activity and breathing rate of a person sitting on the chair [BF14]. This requires to place suitably sized electrodes strategically, at locations that allow getting a good measure of the most significant body parts. Using eight electrodes distributed between seat, backrest and armrests, the sensor values are mapped to postures using two different methods, according to the

specific purpose of the underlying application. The first method directly translates the sensor values to body part positions, while the second uses a machine learning classification.

Direct manipulation of skeleton model

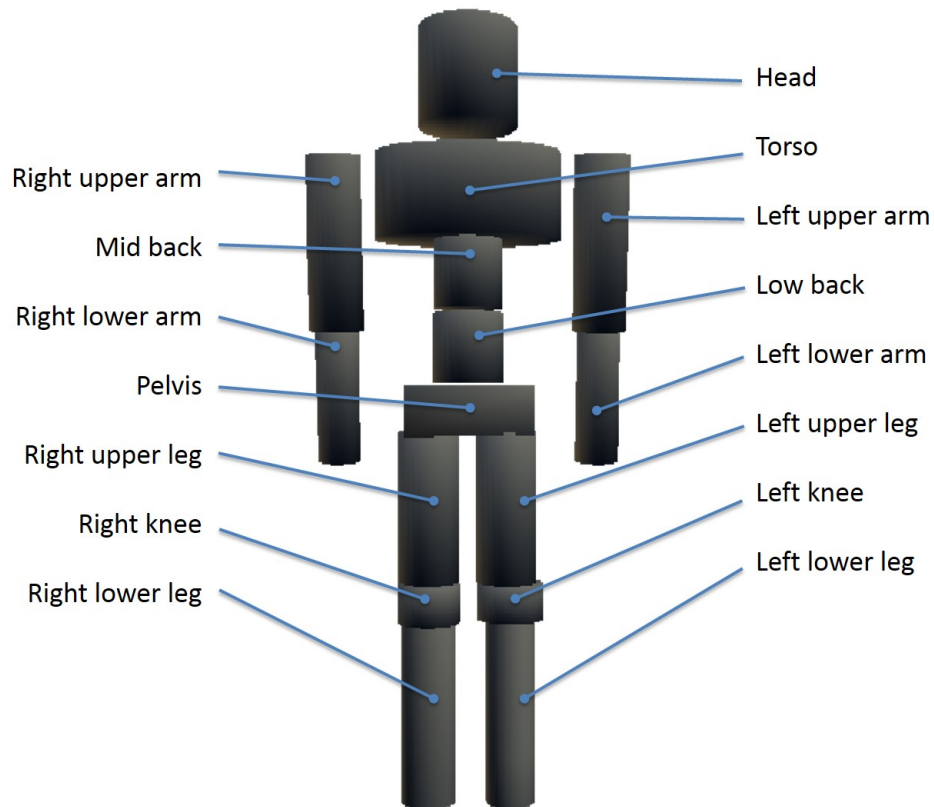


Figure 4.10.: Smartchair skeleton model and associated body parts

In Figure 4.10 the skeleton model can be seen. It is comprised of 15 different groups. As those are combined to each other, the degrees of freedom are significantly reduced to just 10. Sensors close to the different body parts will modify the associated parts of the model. As an example, the distance of the left lower arm group is determined by the sensor in the left armrest. However, as the groups are connected, this will also modify the orientation of the left upper arm. Accordingly, if the sensor behind in the lower backrest detects a receding lower back, the position and orientation of the other back groups, head and arms are also modified. Thus, the fitting method attempts to place the different groups of the model according to current sensor readings. A collection of rules is used that follows the flowchart outlined in Figure 4.11.

The calculation of the different steps varies and is outlined shortly in the following enumeration:

1. *Check sensor activity* uses a threshold of the different sensors to determine the presence of any objects in detection distance
2. *Calculate proximity* use a second thresholds of seat sensors to determine sitting or non-sitting status

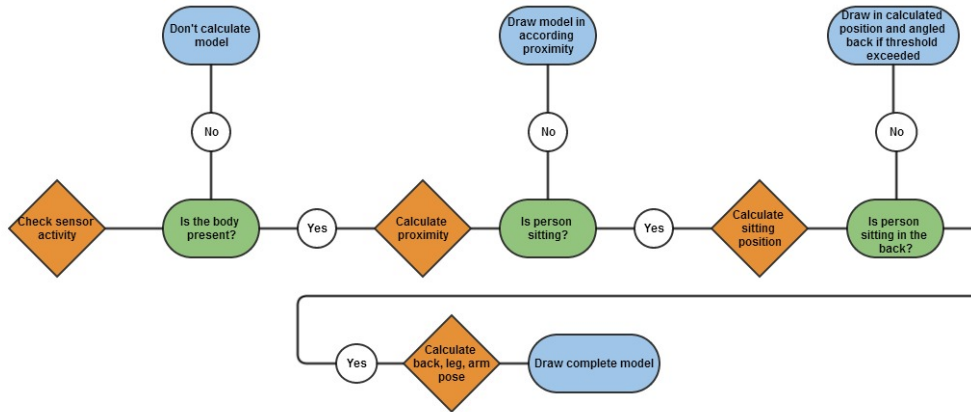


Figure 4.11.: Flowchart of the model fitting process of the grouped skeleton parts and performed calculations

3. *Calculate sitting position* use a weighted average of the seat area sensors to determine position on seat
4. *Calculate back, leg, arm pose* calculate back pose by determining distance from back to backrest based on sensor values. Raise leg according to proximity to front seat sensor. Calculate arms based on proximity to arm rests.

This method allows a fine-grained fitting of the model to the sensor values that closely resembles the pictures of a person moving in the chair. A number of poses and the accordingly fitted 3D models using this method can be seen in Figure 4.12.

Using a fitting method that allows to detect fine movements enables a number of unique applications. One example is to track a set of different exercises that can be performed on an office chair, in order to prevent future back problems. Following an exercise routine can significantly reduce the indicative risks [RAD*09]. Some of the common exercises can be tracked for accuracy and number of repetitions using the capacitive proximity sensors in the chair.

A limitation of the above method are the fixed threshold levels that are set based on experience. Either a calibration method can be used that personalizes the different thresholds, or another form of classification. This should be based on training with a larger variety of different body volumes. It is suitable to use machine learning methods for this task, as they are designed to learn from a larger set of training data and provide numerous classification methods. However, this leads only to a discrete set of different postures, limiting some potential applications. One method is described in the following section.

SVM classification

Support vector machines (SVM) are a supervised learning method that is primarily used for linear classification of n-dimensional features [HDO*98]. They are clustering data by calculating a hyperplane from training data that maximizes the distance from the closest features. A fast learning method is using sequential minimal optimization and was proposed by Platt [Pla99]. The algorithm requires normalized values. A dynamic normalization algorithm constantly analyzes the sensor data for minimum and maximum values and accordingly calculates the normalized value. There is a variety of different software frameworks for machine learning that support training

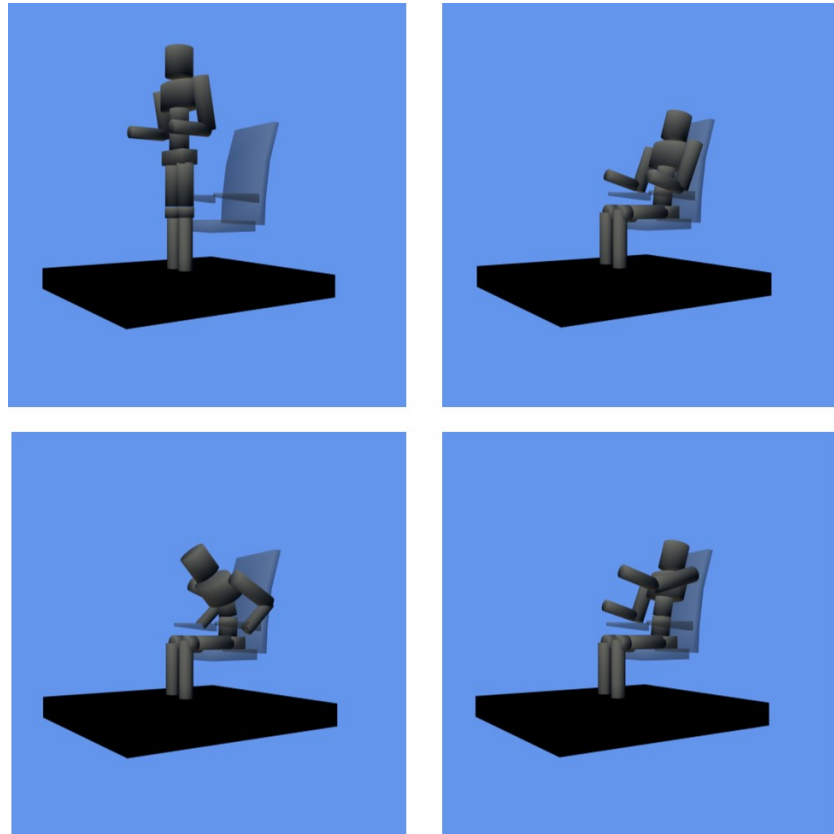


Figure 4.12.: Screenshots of the Capacitive Chair application showing different poses of the skeleton model

and recall of SVMs, thus there is no need for reimplementing these methods. As there is an implicit weighting of features according to significance, there is no need to pre-process or weigh the sensor data.

The training data is collected from a set of persons that have a significant variance in body shapes in both height and girth. SVMs support an arbitrary number of different groups for classification. However, the number of significant poses on a chair is limited. The Global Posture Study by office furniture manufacturer Steelcase Inc. analyzes the most common poses with a focus on information consumption from modern technical devices, such as smart phones or tablets [Ste13]. The different postures are shown in Figure 4.13. A capacitive office chair should be able to distinguish most of these poses if training data has been collected from a sufficiently large number of suitable candidates. Additionally, using sensors clearly positioned on a certain side of the chair, e.g. the armrest, it is possible to associate directional varieties of the asymmetric postures.

4.2.3. Heterogeneous sensor systems

Heterogeneous sensor systems have been an active research topic in the last decades. They can be described as a collection of sensors in the same domain that are measuring different physical properties, or the same properties using different methods [BJ98]. The majority of research is aimed to enable a self-organization of networks of

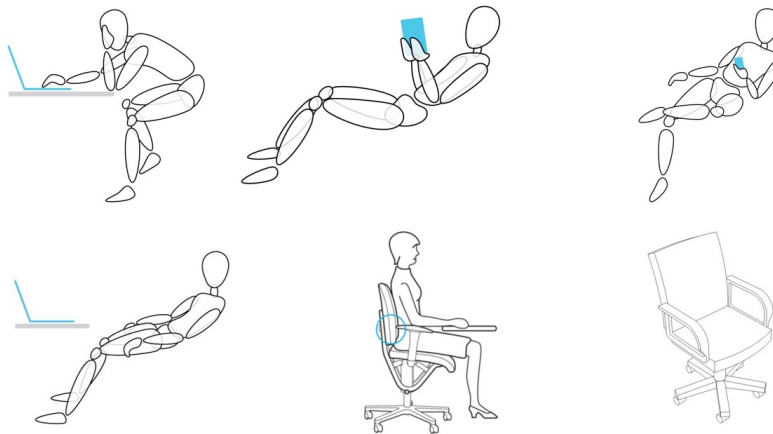


Figure 4.13.: Selected set of postures from Global Posture study and own gestures. *From top left to bottom right:* The strunch, the draw, the smart lean, the take it in, upright, no person (first four taken from [Ste13])

heterogeneous sensors, or combine the data of heterogeneous sensors in a meaningful fashion. For capacitive proximity sensors another factor of heterogeneity has to be considered - the vastly flexible shape and material of the electrodes. This allows to create ensembles of capacitive sensors in a single domain that serve different purposes, e.g. large electrode sensors that detect the presence of a body over a longer distance, combined with small electrode sensors in a specific area to detect touches. The more traditional approach combines the data generated by different categories of sensors in a meaningful fashion, in order to create higher level information that would not be available otherwise. One example of this approach is the previously presented LaZMouse by Smith that added palm proximity sensing to a typical computer mouse [Smi99].

In this section I will present two different contributions. The first is a concept for a heterogeneous capacitive sensor array that combines large electrode sensors to detect the presence and configuration of limbs and numerous densely packed small electrodes enabling gestural finger interaction. The second system aims at alleviating the inability of capacitive proximity sensors to clearly identify touches if the electrodes are at a distance from the touch surface. Data that is acquired from a microphone listening for the acoustic effects of a finger touching the surface are combined with capacitive proximity sensor data. This allows to combine mid-air interaction and touch recognition.

4.2.3.1. Heterogeneous capacitive arrays

An often-cited issue of mid-air interaction systems is discriminating between an intended gesture and arbitrary movements in the detection area [HPGK94]. While traditionally used for full-body gesture systems, this is also true for capacitive proximity sensors that provide 3D tracking. If the interactive zones are to be integrated in design features that serve multiple purposes, there needs to be methods that distinguish between typical use and intended interaction. In a project with students Sönke Schmidt and Stephan Neumann I developed a concept for a car armrest equipped with a capacitive finger interaction system [BNS*14]. A regular car armrest is augmented in a way that integrates the capacitive sensors in an invisible fashion. In this case it is necessary to clearly distinguish between intended control gestures by the driver or if he is just resting the arm. The idea is to use the status of both arm and hand to identify the intention. The system is using a heterogeneous combination of

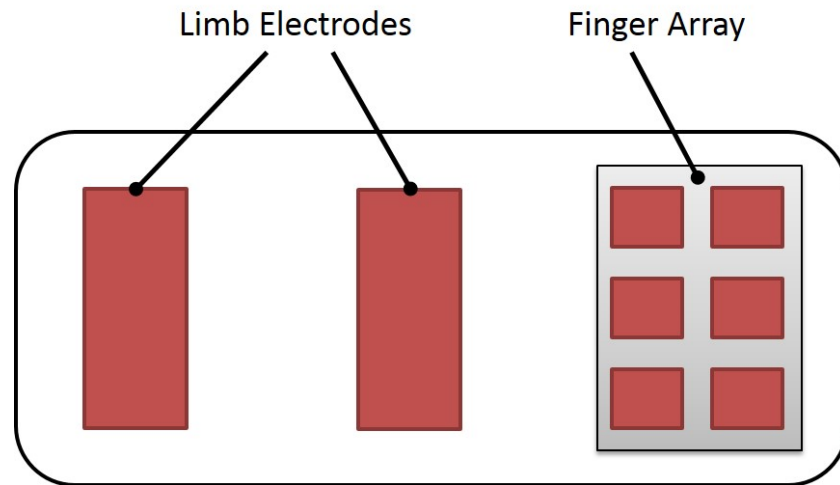


Figure 4.14.: Heterogeneous sensor array for limb detection and finger tracking

capacitive proximity sensors. Sensors in the middle and back of the armrest are used to detect the current status of the arm, while an array of smaller electrodes in the front can track a variety of finger gestures. The setup is shown in Figure 4.14.

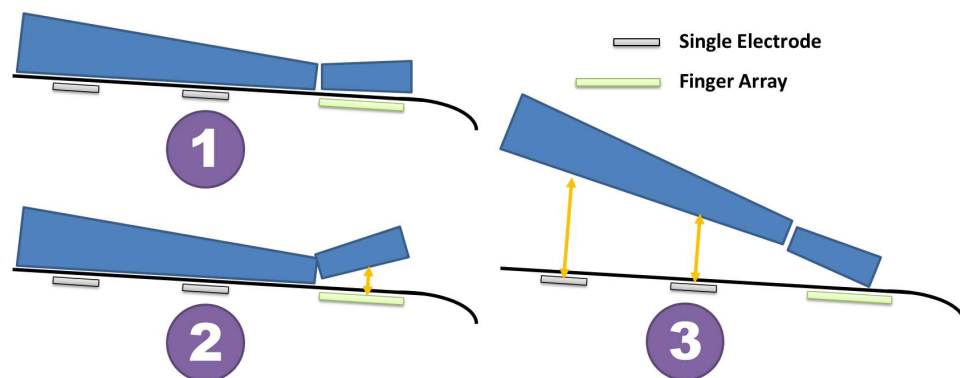


Figure 4.15.: Arm model and detection of posture based on distances to two sensors and finger array for resting position (1), hand raised position (2) and arm raised position (3)

In Figure 4.15 three different positions are shown that arm and hand can have on the armrest.

1. Arm and hand are in resting position with both close to the surface
2. Arm resting on the back and the hand hovering in proximity of the front area
3. Arm raised position and fingers touching the front of the armrest

The latter two positions are suitable for finger-based gestural interaction as they can be moved freely. The system should therefore be able to distinguish between the three different positions and consider either position (2) or (3) or both as intended interaction. Different interaction patterns have to be chosen for both active positions.

Regarding the arm raised position the person will typically want to interact using familiar touch gestures. In the hand raised position it is necessary to track gestures that are performed in the air. In both cases we assume that a single finger is used.

The data processing of the system therefore requires three distinct steps. At first the three specified limb postures have to be identified. Afterwards, the position of the fingers in or above the interaction area have to be calculated, and finally a time-series analysis of subsequent positions has to be performed, in order to infer different gestures. The two distinct arm sensors are able to determine single distance values. In addition the aggregated data of the finger detection array in the front is used to detect a third distance value. To map the different postures, a set of thresholds is used that determine if arm or hand are touching the armrest surface, or if they are hovering above it. Be r_b the value of the sensor in the back, r_m the value of the sensor in the middle and r_f the aggregated value of the sensor values in the front $r_{f,i}$. Using three threshold levels t_b , t_m and t_f the different postures $p \in 1, 2, 3$ can be determined as follows

$$r_f = \sum_{i=0}^n r_{f,i} \quad p = \begin{cases} 1, & r_b \geq t_b \wedge r_m \geq t_m \wedge r_f \geq t_f \\ 2, & r_b \geq t_b \wedge r_m \geq t_m \wedge r_f < t_f \\ 3, & r_b < t_b \wedge r_m < t_m \wedge r_f \geq t_f \end{cases} \quad (4.8)$$

As we are acquiring sensor data proportional to distance, it is also possible to calculate orientation angles of the arm and use it as input. However, for now this was not followed up any further. The calculation of the finger position in three dimensions is adapted from the previously introduced method for sparse capacitive arrays to track objects in three dimensions. It uses a combination of weighted average for planar location and stepwise linear interpolation to determine the height.

To classify the gestures we are using points from a distinct start to a distinct stop. In case of the free-air interaction this is determined by the finger moving in a constrained part of the interactive area. In case of the touch interaction it is determined by a finger starting and stopping to move, while exceeding a threshold indicating touch. The points between start and stop position are normalized and a SVM classifier is used to detect the trained gestures. There are distinct classifiers for the two different methods that are triggered according to the selected interaction pattern. The SVM is trained using the sequential minimal optimization method by Platt [Pla99].

4.2.3.2. Heterogeneous sensor fusion

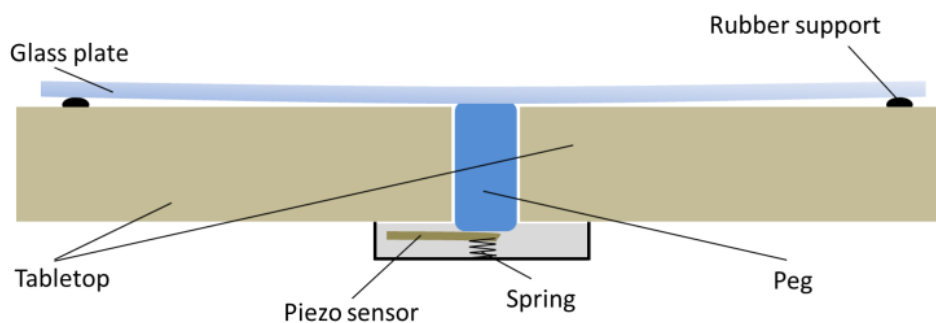


Figure 4.16.: Suspended peg knock detection system

As previously mentioned, capacitive proximity sensors inherently lack the ability to detect touches if the electrode is at a distance from the touch surface. This may be overcome easily in systems that are tuned to work

in close proximity, e.g. SmartSkin [Rek02]. Another option is adding a different sensor category specifically for detecting touches. The pressure applied to the surface can be detected by a suitable sensor, e.g. by using a surface suspended on sensors, such as the weight-aware dining table by Chang et al. [CLC*06]. However, this requires a specific construction and can't be easily fitted to existing systems. Additionally, there are restrictions to the potential rigidity. A different effect of surface touches is the acoustic response. A microphone connected to the surface can detect a variety of different touch events. Harrison et al. presented a system that detects varieties of fingernail scratches performed on a rough surface [HH08]. The system was later extended to detect impact events by different parts of a finger or pens [HSH11].

In collaboration with Sebastian Zander-Walz and Stefan Krepp I have tested different methods to combine sensors for dedicated touch detection and capacitive hand tracking. They are integrated into an existing living room table, thus providing a potential to retrofit existing pieces of furniture [BZWK*14]. A first idea was to use a single piezo sensor to detect vibration of the glass plate that covers a living room table. The concept is shown in Figure 4.16. This first system was prone to outside influence and had difficulties detecting multiple touch events. The second iteration used a microphone system, similar to the concept provided by Harrison. However, an extension to their approach is possible that not only classifies impact events, but additionally double impacts and swipe gestures that are performed over a longer time scale. It will be described how both effects can be evaluated simultaneously and the required modifications to the previous methods. The basic idea is to analyze the acquired acoustic signals in the frequency domain and use a classification method based on a number of selected features related to frequency. The supported gestures and the associated FFT profiles are shown in Figure 4.17.

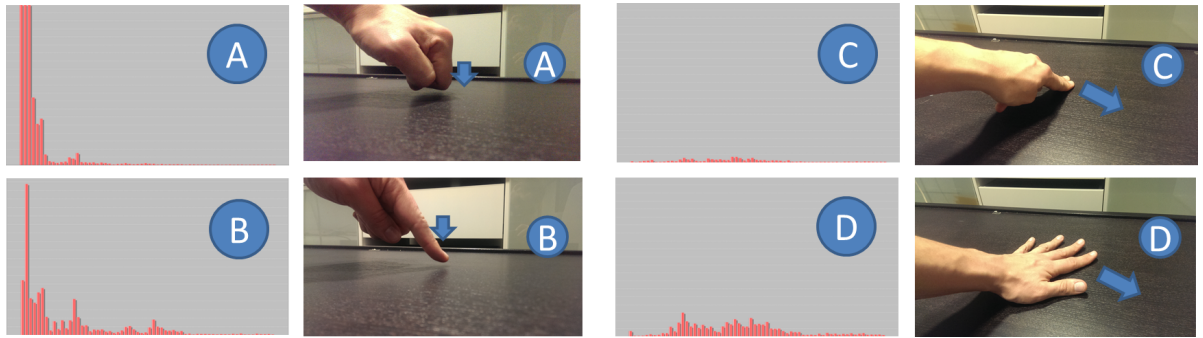


Figure 4.17.: 64 sample FFTs and photo for a knock event (A), a finger tap (B), a finger swipe (C) and a hand swipe (D)

Features At first, the audio signal is acquired using a 96kHz sampling rate. A Hanning-type sliding window of 4096 samples (43ms) with 256 sample overlap is used, resulting in a sampling rate of 375Hz. A variety of features is used. A FFT over 4096 samples is performed, leading to 2048 magnitude values. The first 512 of those cover the most significant frequency range up to 12kHz (512 features). This process is repeated for a downsampled FFT of 64 values (64 features). For both mean value, standard deviation and index of the highest value are collected (6 features). Additional features are the centroid (weighted mean of the present frequencies) and two time-domain features, the RMS power (root mean square - average magnitude within the current frequency band) and the number of zero crossings (3 features). Overall, this leads to 585 features used for classification. To detect double impacts a heuristic is used that searches for two uniform impact events in a short time frame. The window for double events has to be chosen carefully to be suitable for most users. In a study times between 112ms and

461ms for double events were collected. With some additional error margin 533ms was chosen as window for our system.

Swipe events The swipe detection uses an additional feature set, which is calculated from a sliding window over the last ten feature vectors. A downsampled FFT of eight values is used and we calculate the average and the standard deviation in the sliding window. The result is two vectors AVG_{FFT8} and $STDEV_{FFT8}$, both containing eight values. A swipe condition is satisfied if AVG_{FFT8} exceeds a predefined threshold and $STDEV_{FFT8}$ does not exceed the corresponding standard deviation threshold. Heavy impacts may meet the average conditions but won't satisfy the standard deviation conditions, as the majority of the signal is early in the window. This effect is shown in Figure 5, with impact events and their low-frequency peaks on the left and swipe events and their fairly constant value on the right. Finger swipes are distinguished from hand swipes by looking at AVG_{FFT8} . In order to identify a swipe gesture, a number of subsequent positive classifications have to occur. Ten subsequent classifications are identified as a swipe gesture. They indicate a constant movement of 27ms or longer.

Classification Two different classifiers for impact and swipe events are required. These are both trained SMO classifiers comprised of a support vector machine and a normalized polynomial kernel [HDO*98]. At first there is a test for swipe events, due to their longer duration. If no swipe was detected the features are pushed into the impact event classifier. Overall we can distinguish seven events. The impacts, single knock, double knock, single tap, double tap and stomp. The swipe events are hand swipe and finger swipe. The duration of the classification process is about 75ms (Notebook with Core i7-4500U, 6GB RAM), which makes it applicable for real-time application.

4.2.4. Image-based processing

Their ability to detect changes in the electric field over a distance has led to capacitive proximity being regarded as similar to cameras. Smith et al. consequently called their approach electric field imaging, as particularly shunt mode measurements and their constrained electric fields allow applying certain image processing methods, e.g. tomography [Smi99]. They were critical of using similar methods for shunt mode, noting the following statement.

Loading mode measurements can be likened to images formed without a lens, since only one "end" of each field line is constrained by the measurement. [SWD*98]

Nonetheless, loading mode has certain advantages, particularly if all electrodes are in a single plane and we would like to have a higher sensitivity at a distance from the plane it is advantageous if there is no receiving potential nearby. One example for this planar electrode setup is large area gesture interaction devices, e.g. a table that is able to track the position of arms and hands in three dimensions. There is a plethora of image-based object detection and tracking algorithms that can be also used for capacitive proximity sensor data processing. There is a short process that I propose to realize this arm and hand tracking that includes some general steps that can be used to identify a variety of objects [BZWK*14]. The process is distinguished into four distinct steps:

1. Creating a capacitive image from the acquired sensor data
2. Apply a feature-preserving image upscaling method
3. Find the contours of the present objects according to pixel values
4. Analyze the image moments of the contour areas and fit human arms



Figure 4.18.: Pixel array mapped from sensor values

4.2.4.1. Creating a capacitive image

The most challenging aspect of the first step is the low resolution of a reconstructed image. In order to achieve a mid-range distance resolution that allows detecting objects within 30cm to 40cm, it is necessary to use electrodes that are sufficiently large. Thus, this method uses an array of just 6x4 sensors. The result is a capacitive image of just 24 pixels. Typically the sensor values are an integer in a range between 0 and 15000. Accordingly, it is possible to create a single-channel image with a channel depth of two bytes. In this case a linear mapping of sensor values to pixel intensities is used. An exemplary result image of this mapping is shown in 4.18 (with enlarged pixels). In this format it is difficult to gather information about the exact position of the arms and thus further processing has to be applied before calculating the contours and fitting arm objects.

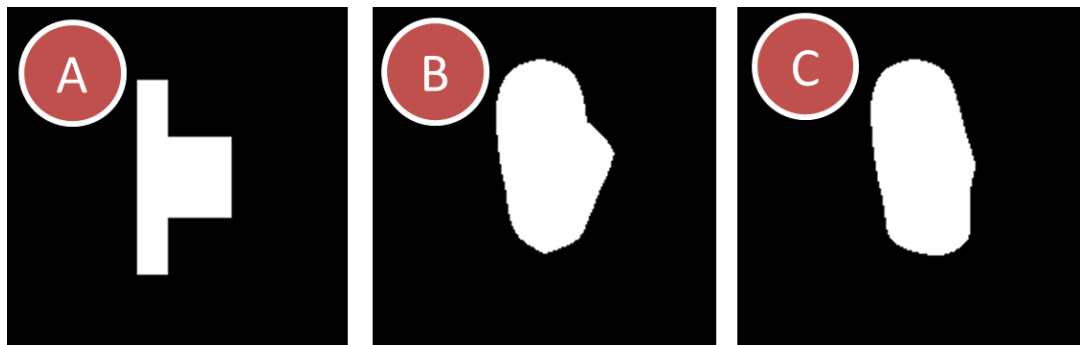


Figure 4.19.: Effect of different upscaling methods on shape, (A) nearest neighbor, (B) bicubic, (C) bilinear, (D) Lanczos4 - shown as thresholded binary images (pixel intensity > 30)

4.2.4.2. Upscaling of the capacitive image

The next step is to enlarge the image using a feature-preserving scaling method. As all sensors are prone to environmental noise, we apply thresholding based on the pixel intensities before looking for contours. The result is an enlarged binary image of black and white pixels. There are two prerequisites to the image scaling method. First, it should be invariant towards rotation, as the resulting contour for an angled arm should not differ from

a straight arm contour. Second, the method should preserve the shape of the contour edges, in order to improve the quality of the features derived from the contour for later processing. The category of windowed sinc filters supports both prerequisites [Get11]. For our system the variety of a Lanczos4 kernel was chosen. While this interpolation method is processing intensive it is reasonable in this context due to the small size of the base image.

4.2.4.3. Finding the contours

In order to find the objects in the capacitive image, we apply a thresholding method to separate the objects from the background. The result is a binary black and white image. The contours are defined as the borders between black and white regions. A comparison between three different upscaling methods and the resulting contours after thresholding, is shown in Figure 4.19. Nearest neighbor filtering does not capture the shape of the arm very well. Bilinear upscaling provides better results. However, there are artefacts due to rotation of the object. Lanczos4 provides a good shape approach, even when rotated.

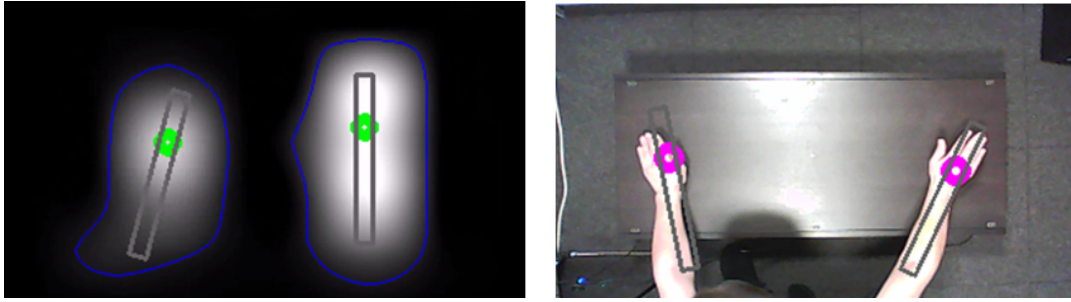


Figure 4.20.: *Left*: Capacitive image of two arms, detected palm position and arm orientation. *Right*: Overhead camera picture of the scene overlaid with live arm and palm reconstruction for two arms

4.2.4.4. Palm and arm fitting

The last step of identifying and tracking the arms is to fit the position and orientation of the palms and arm into the acquired object contours. For this task the image moments within the contours are analyzed. These are certain particular weighted averages of pixel intensities, or a function thereof [Hu62]. They can be calculated using the following equation, whereas j and i define the order and $I(x, y)$ is the pixel intensity at a given position. This can be used to calculate the center point (\bar{x}, \bar{y}) , leading to the central moments mu_{ji} that are required to determine the orientation of the contour as angle γ .

$$m_{ji} = \sum_{(x,y)} I(x,y) x^j y^i \quad (4.9)$$

$$\bar{x} = \frac{m_{10}}{m_{00}}, \bar{y} = \frac{m_{01}}{m_{00}} \quad (4.10)$$

$$mu_{ji} = \sum_{(x,y)} I(x,y) (x - \bar{x})^j (y - \bar{y})^i \quad (4.11)$$

$$\gamma = 0.5 \cdot \arctan \frac{2 \cdot mu_{11}}{mu_{20} - mu_{02}} \quad (4.12)$$

The center point and orientation are used to calculate the estimated position of the palm of the hands. These points are the basis for the subsequent gesture recognition. Additionally, separate Kalman filters are integrated to smooth the different palm positions and arm orientation. The resulting arm reconstruction and the actual arm position in a photo are shown in Figure 4.20. For this image a simple webcam was installed above the table and its output registered to the table position. The arm reconstruction allows us to determine the orientation of the arm and the direction from which the table is. Another potential use of the arm orientation is to improve the merging of two hands. While the system can't distinguish from a single sensor if one hand is close or two hands are further away, we can use the presence of two arms to identify the overall number of objects in the detection range.

4.2.4.5. Intensity-based elevation estimate

A distinct challenge of the capacitive hand tracking is the considerable directional difference in available resolution. While we can use the presented image analysis to track the planar position of the arms over the whole table area of 80cm width and 50cm depth, estimating the elevation of the arm above the table is restricted by the proximity range of the single sensor. Typically the achievable range has a maximum of approximately 35cm, depending on environmental conditions. In a plate capacitor system the distance d is proportional according to size of the plates A and resulting capacitance C . Due to the linear mapping of sensor capacitance measurements to pixel intensities I we can use the image moment within a contour S as estimate of the actual capacitance, and calculate the elevation e according to the following equations:

$$d \propto \frac{C}{A} \qquad S \propto \frac{m_{00}}{\int S} \qquad (4.13)$$

The same thresholds discussed in the contour retrieval phase apply to this step, thus leading to discarding objects at a larger distance that are difficult to detect. Starting from this threshold the resulting elevation is normalized according to a maximum threshold for m_{00} that denotes a very close object (such as touch). The actual touch recognition is performed using acoustic methods. As previously explained the sensors are prone to environmental influences, thus this just allows to get an estimate of the actual elevation and no absolute distance value.

4.2.5. Physiological signals in frequency- and time-domain

All underlying physiological processes are based on the transmission of electrical signals, resulting in a number of electrical measurements in the medical domain, such as EEG or ECG. However, these systems rely on a close contact to the skin, in order to measure the electrical signals produced inside the body. Remote sensors have to use a different strategy - the effect that many physiological effects are accompanied by muscle movements that result in temporary changes of posture or volume. Capacitive proximity sensors can be used to measure various physiological parameters that are related to movement of different body parts. This may include internal organs, most notably the heart. Cheng et al. have presented a system that allows measuring motions and shape changes of body parts using capacitive sensors embedded in garment [CAL10]. They were able to detect swallowing and breathing rate. One example for an industrial application is non-contact electrocardiogram (ECG) sensing in cars, intended to detect drowsiness in drivers. Using three electrodes it is possible to detect the heart rate or even acquire a full ECG through various layers of clothing [Ple12]. MacLachlan presented a system that detects the respiratory rate of a person lying on a bed from a distance of up to 50cm using a single electrode

and a highly sensitive sensing method based on spread spectrum methods that are commonly used in wireless communication [Mac04].

In this section I present two contributions. The first is an approach to detect the respiratory rate using an analysis in the frequency domain, based on readings by a single capacitive sensor applied close to the chest. The second system uses movement sensing of a person lying in a bed to identify sleep phases.

4.2.5.1. Respiratory rate

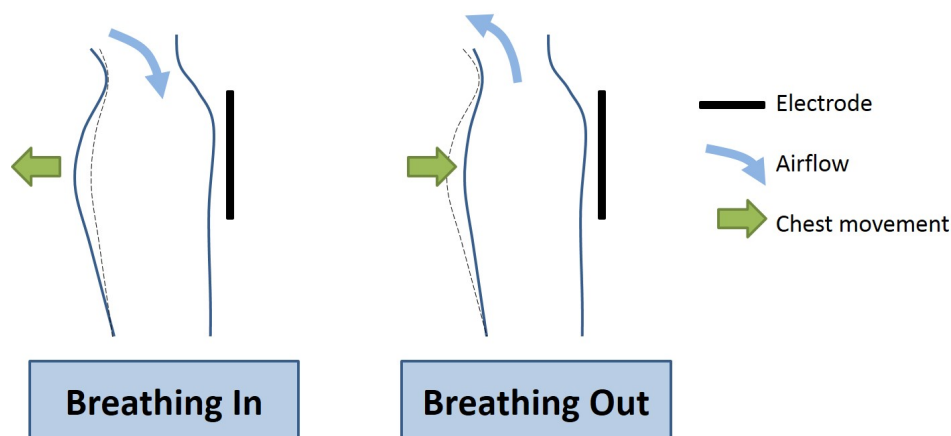


Figure 4.21.: Chest movement when breathing in and out

The volume changes of the chest while breathing have been a topic of research for a long time [Wad54]. If the body of a person is not moving and can be considered at a static distance from a capacitive proximity sensor, the chest movement should translate into a periodically changing sensor value. This system has been developed in cooperation with Sebastian Frank and exemplified using a chair [BF14]. The breathing rate detection is operating on a single electrode that is placed close to the chest. The basic concept is shown in Figure 4.21. The surface of the electrode is large and close to the surface. Therefore, it is able to pick up the chest movement. Two different methods of data processing are used and fused to get the final breathing rate. Using a fast Fourier transformation the signal is transformed into the frequency space. We are looking for significant signal portions in frequency areas that can be associated to breathing, between 0.1Hz and 3Hz.

The above Figure 4.22 shows an example of the sensor data curve generated by the conductive thread sensor behind the back of a person. The chest movement is clearly visible as sinusoidal oscillation of the sensor value. If there is a sufficiently stable baseline, the zero-crossings can be calculated. However, as this can't be guaranteed in all cases an adaptive baseline should be used that is reconfigured according to changing states of the sitting person.

4.2.5.2. Sleep phase recognition

Using movement data to detect sleep phases is an unobtrusive method of sleep monitoring. Studies have shown that the magnitude of movement is typically associated to the following phases in decreasing order: wake, stage 1, REM, stage 2, stage 3 [WFS83]. Another method is distinguishing between awake phase, active sleep and

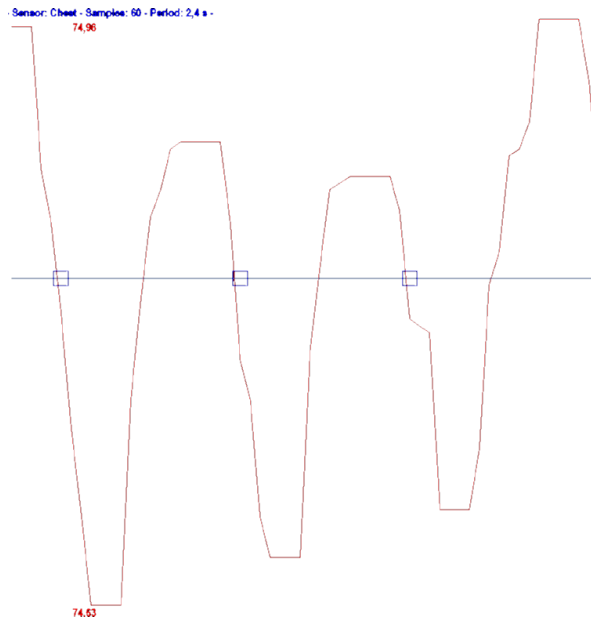


Figure 4.22.: Sinusoidal curve caused by chest movement above a capacitive sensor placed close to the chest

quiet sleep and takes into account the order of those phases. This information allows to correlate the actual sleep phases with good certainty [SL86]. Together with Maxim Djakow and Alexander Marinc I have been working on a system to distinguish sleep phases based on movement data acquired from capacitive proximity sensors [DBM14]. A typical distribution of sleep phases throughout the night is shown in Figure 4.23. It can be easily seen that the sleep is distributed into different cycles, whereas the sleeping person is moving through the different sleep phases until having a REM phase and then going back to deep sleep. If the only available data is body movements it is becoming more difficult to reliably determine the sleep phase.

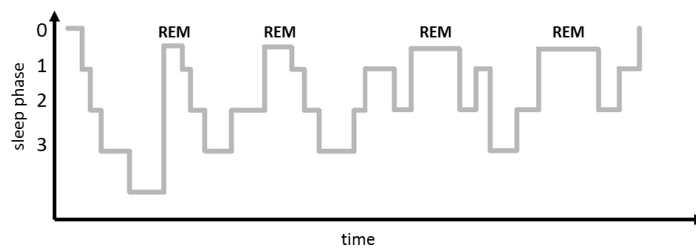


Figure 4.23.: Example of human sleep phases throughout the night

The most reliable way to track sleep phases is by using an electroencephalography (EEG); that is measuring the electrical activity of the brain by placing electrodes on the scalp. Various different types of neural oscillations can be distinguished - the most important for sleep phase detection are alpha waves, theta waves, delta waves and sleep spindles. The American Academy of Sleep Medicine (AASM) distinguishes three different phases of non-rapid eye movement sleep (NREM) and REM phase [Sch08].

- Stage 1 - occurs mostly in the beginning of sleep. It has slow eye movement, alpha waves disappear and the theta wave appears.
- Stage 2 - dreaming is very rare and no eye movement occurs. The sleeper is quite easily awakened. EEG recordings have a tendency for characteristic "sleep spindles"
- Stage 3 - was previously divided into stages 3 and 4. It is slow-wave sleep (SWS) or deep sleep. Stage 3 used to be the transition between stages 2 and 4 where delta waves began to occur, while delta waves are dominant in stage 4.
- REM sleep - is a phase of sleep characterized by random and rapid movement of the eyes. It is considered the lightest phase of sleep and occurs all through the night but gets longer close to morning.

Capacitive proximity sensors enable us to detect the presence of suitable objects and their relative proximity to the electrode. Consequently, a moving object will cause a change of sensor values. If these data deviations from an array of sensors are aggregated, the result is a reliable measure of objects moving above the electrodes. In the case of our system, it is assumed that there is a limited number of persons moving on top of the sensors. Thus, it is possible to associate the sensor values to movement. In the following I will present a suitable method to achieve a reliable detection of the movements of a sleeping person, whereas I am following a similar approach as Salmi and Leinonen [SL86]. At any given time t a set of the latest values of all n sensors can be stored as a tuple in the following form:

$$\vec{s}_t = \begin{pmatrix} s_{1t} \\ s_{2t} \\ \vdots \\ s_{nt} \end{pmatrix} \quad (4.14)$$

As capacitive proximity sensors are particularly susceptible to external influences, such as temperature, humidity and other electric fields it is necessary to apply filtering on the sensor values. A suitable candidate is a median filter - a low-pass filter method that selects the median object of a sorted set of values, thus discarding outliers and strongly deviating values. This is particularly suited if transmission errors may occur. If a person is moving on the bed the value of all sensors in detection distance of the moved body parts will change accordingly, the most relevant example in our case being a person moving in its sleep. We can generate a measure of movement intensity by comparing the values at time t with those at time $t - 1$ resulting in:

$$\vec{d}_t = |\vec{s}_t - \vec{s}_{t-1}| = \begin{pmatrix} |s_{1t} - s_{1t-1}| \\ |s_{2t} - s_{2t-1}| \\ \vdots \\ |s_{nt} - s_{nt-1}| \end{pmatrix} \quad (4.15)$$

In subsequent calculations, I will use \vec{d}_t as combined measurement. For distinguishing between wake, active sleep and quiet sleep it is solely interesting to get the most intense movement. Thus, we are testing for the largest value over a set of m samples, generating the value b_t .

$$b_t = \max(\vec{d}_t 1, d_t 2, \dots, d_t m) \quad (4.16)$$

The value b_t is affected by changes in the speed of movement. Therefor as a final step we generate a centered average value of order $2q - 1$:

$$\bar{b}_t = \frac{1}{2q-1} \sum_{i=-1}^q b_{t-i} \quad (4.17)$$

The resulting value \overline{b}_t allows quantifying the intensity of movements over a given period. In order to extract an actual body movement from this value, it is necessary to quantify a threshold $s(t)$ that is determined by the average of q previous values of \overline{b}_t multiplied with a factor f that has to be evaluated individually for each configuration of bed and sensors. This threshold $s(t)$ allows identifying a movement m at any time t . This behavior is denoted in the following equations:

$$s(t) = \left(\frac{1}{q} \sum_{i=1}^{q+1} \overline{b}_{t-i} \right) \cdot f \quad (4.18)$$

$$m_t = \begin{cases} 1, & \text{if } \overline{b}_t > s(t) > \overline{b}_{t-1} \\ 0, & \text{else} \end{cases} \quad (4.19)$$

As previously mentioned, it is difficult to determine sleep phases solely by monitoring the movement. Instead following the example of Salmi and Leinonen three different phases are distinguished - wake, active sleep and quiet sleep [SL86]. These are determined by dividing the sleep time into a three-minute epochs e_{i_a} and qualify these as active or quiet by counting the number of movements occurring in those intervals and comparing it to the average amount of movements in all epochs \overline{e}_a determined by the following equations:

$$e_{i_a} = \sum_{e_{i_{start}}}^{e_{i_{end}}} m_i \quad \overline{e}_a = \frac{1}{n} \sum_{i=0}^n e_{i_a} \quad (4.20)$$

In consequence, the status of any epoch is determined using this final equation:

$$e_{i_a} = \begin{cases} \text{active}, & \text{if } e_{i_a} > \overline{e}_a \\ \text{quiet}, & \text{if } e_{i_a} \leq \overline{e}_a \end{cases} \quad (4.21)$$

These active and quiet periods can be semi-autonomously interpreted by humans, in order to determine the actual sleep phases. For example initial activity for 20 to 40 minutes followed by a quiet period can be attributed to a person falling asleep. Following quiet phases are a good indicator for deep sleep phases.

4.3. Application prototypes

In the last few years I have designed a number of different prototypes using capacitive proximity sensors in various usage scenarios within smart environments that were realized with the help of numerous students and colleagues. They tackle specific application domains and implement one or more of the data processing methods that have been specified in the previous section. A short overview can be found in Table 4.3. During the next few pages I will describe in detail how the prototypes have been created, how they implement the different data processing methods and outline the results of any technical evaluation and usability study that has been performed.

Table 4.3.: Overview of developed capacitive proximity sensing prototypes

Name	Description	Application Areas	Measuring Layout	Data Processing
CapFloor	Capacitive system for indoor localization and fall detection based on electrode grid below the floor.	Indoor Localization	Loading mode, variable number of sensors based on area size	Binary activity association and using geometry for positioning. Monitoring of overall value for falls.
Smart Bed	Capacitive sensors placed below mattress able to determine sleeping postures and breathing rate.	Smart Appliances, Physiological Sensing	Loading mode, four sensors on each side of bed	Posture fitting using a static model. Fourier analysis for breathing rate recognition.
The Capacitive Chair	Office chair equipped with capacitive sensors to distinguish different typical work postures and stress levels.	Smart Appliances, Physiological Sensing	Loading mode and shunt mode, eight sensors, heterogeneous sensing capabilities	Model fitting using a dynamic model. Fourier analysis for breathing rate detection. Posture recognition using machine learning.
Active Arm-rest	Heterogeneous system for finger gesture recognition and arm rest identification for automotive applications.	Smart Appliances, Gestural Interaction	Loading mode, heterogeneous layout	Finger positioning using direct calculation. Binary arm presence detection.
MagicBox	Mobile 3D gesture interaction device using an array of electrodes.	Gestural Interaction	Loading mode, six wireless sensor nodes	Geometric detection of hand relative to plane. Adapted mouse methods for gesture recognition.
CapTap	Table capable of detecting 3D gestures and knocks to realize tactile interaction in a living room.	Smart Appliances, Gestural Interaction	Loading mode, 24 capacitive sensors and a single touch detecting microphone	Image-based hand and arm detection. Independent touch detection. Tracking of multiple objects.

4.3.1. CapFloor

CapFloor is a capacitive system for indoor localization and fall detection that is based on a grid array of sensing electrodes placed below a floor covering [BHW12]. A sketch of the system is shown in Figure 4.24. The grid is comprised of insulated wires that are placed orthogonal to each other. Sensors are placed on two sides of the room. Each sensor is performing loading mode measurements. The system is intended to act as both indoor localization system and fall detector. CapFloor can be placed below any non-conductive material, like wood, tiles, or PVC. However, it only works if the distance between the wires and the floor surface is not too far. It can discriminate between a foot being above an electrode or a whole body. Combining this information from various sensors, we are able to get a reliable detection of lying, sitting and standing persons. Using only two sides of

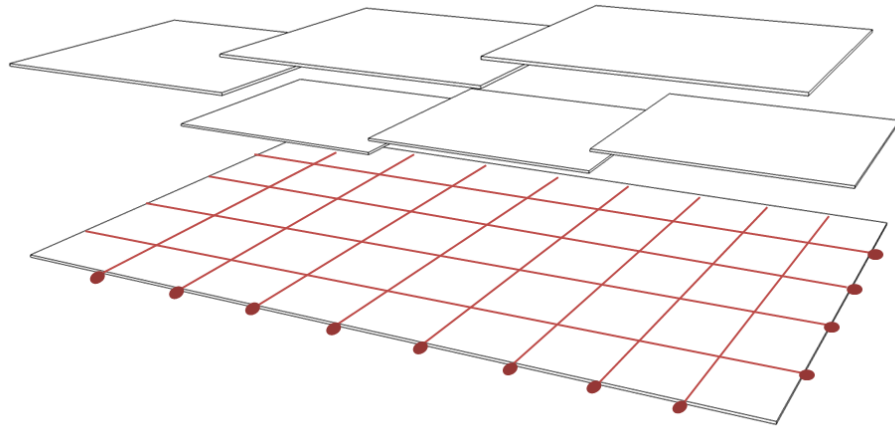


Figure 4.24.: CapFloor sketch - grid layout of electrodes is placed below a floor layer with sensors attached on the sides

the room for sensors it is possible to cut the wires without considerably affecting the signal. This allows easy installation in non-rectangular rooms. Accordingly, CapFloor is able to be used in various application scenarios. Indoor Localization in the home domain can be useful in energy saving and fall prevention by appropriately activating and deactivating the environment lighting. It can also be used in security-restricted areas to detect unauthorized movement. The fall detection should be used in a system that has various levels of escalation. E.g. it is not easy to distinguish between a person doing exercises on a floor and a person that has fallen down. Accordingly the system should query if the person is well and not autonomously call for outside help.

4.3.1.1. Evaluation

The CapFloor system was evaluated in the scope of the Indoor Localization Track of EvAAL 2011, where it participated out of competition [CK12]. In Figure 4.25 we can see a picture of the demonstration setup installed in the living lab using the system integrated into different mats that are placed in the environment. The system was tuned to detect a single person and was able to perform this reasonably in the areas covered. The resolution of the system is strongly depending on the given density of electrode wires. While there is a certain measure of proximity, it is not possible to detect objects that are more than a few centimeters away from the wires. Later iterations of the system are using higher voltages and shunt mode measurements to improve the tracking reliability and enhance the fall detection.



Figure 4.25.: Floor mats with integrated CapFloor system used at the EvAAL 2011 competition [BHW12]

4.3.2. Smart Bed

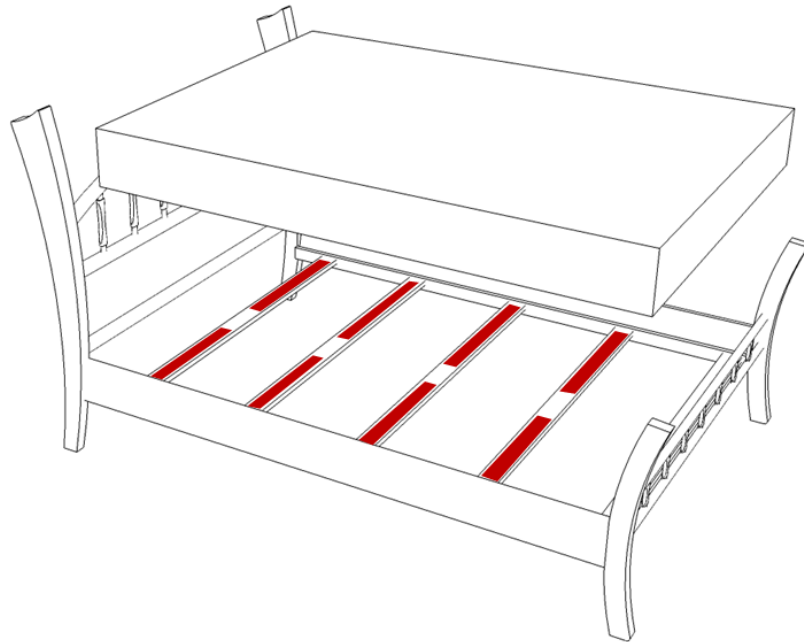


Figure 4.26.: Smart Bed sketch - flexible plate electrode are attached on spring board

The Smart Bed is a regular bed frame that has been equipped with capacitive proximity sensors in order to determine occupation, posture and sleep phases [BH12] [DBM14]. A sketch can be seen in Figure 4.26. The electrodes are comprised of copper foil that is attached to the flexible wooden panels of the slatted frame. This allows the electrodes to be sensitive to both proximity and applied pressure, resulting in a superposed combined sensor value that is considerably higher as opposed to proximity measure on its own. The electrodes are equally distributed, with four being on both sides of the two person bed. The system is able to determine different sitting and lying postures of one or two persons, including less regular lying positions such as lying diagonal or lying orthogonal to the long side of the bed. Using an analysis of the movement gathered by variation in the sensor signal, the sleep phases can be analyzed, similar to accelerometer-based systems that are popular for smartphones [KJJ11].

The Smart Bed can be used for various purposes. A main application is connecting the occupation detection to a home automation system and timer in order to activate ambient lighting if the person is getting up in the night, presumably to find the way to the restroom. Accordingly, in a single person household the lights in unoccupied rooms could be turned off in order to conserve energy. In the domain of personal health the Smart Bed is able to give the user a feedback on sleep quality based on the sleep phase measurement performed in the night. Another potential application is to use the acquired pressure distribution as indicator for back-friendly lying positions that may be harmful over a longer period of time [HB10]. The occupation and posture detection relies on a simplified body model to approximate the pressure distribution and sensor values to a certain posture [BH12].

The different components of the Smart Bed data processing are shown in Figure 4.27. Raw sensor data is distributed to three different modules, the calibration, which is determining the initial parameters for the sensor

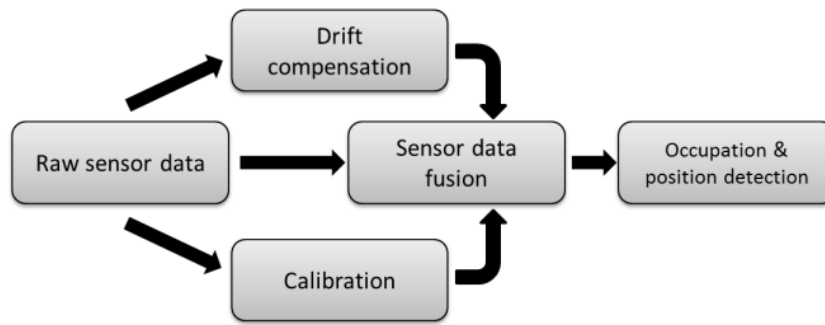


Figure 4.27.: Data processing components [BH12]

data fusion, the drift compensation that alters those parameters according to long term trends and finally the sensor data fusion module that processes the data and does feed it to the occupation & position detection. Both calibration and drift compensation follow the previously presented model [BH12].

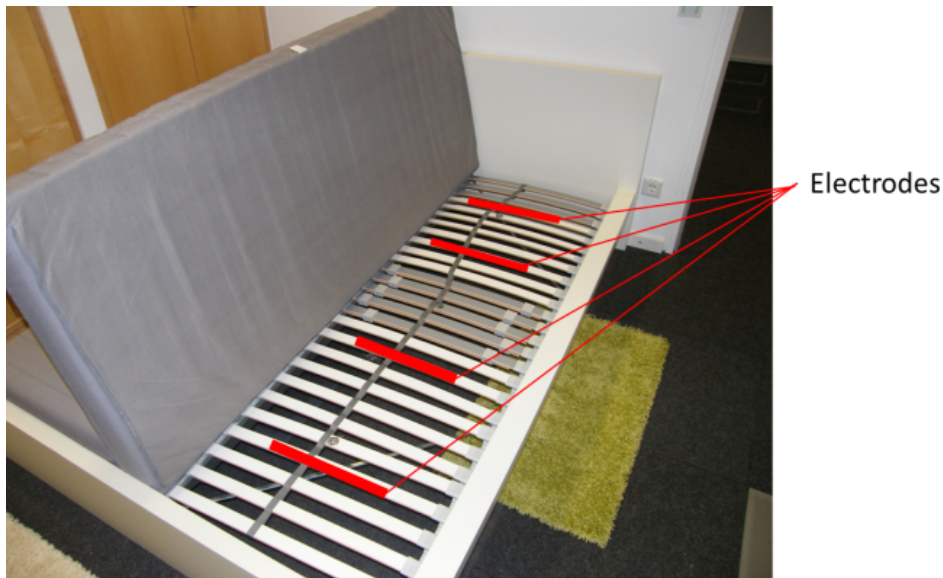


Figure 4.28.: Electrodes and sensors hidden below mattress of Smart Bed [BH12]

The system prototype is shown in Figure 4.28. The positions of the electrodes on the slatted frame are indicated in red. The picture only shows one side of the bed. The same electrode positions are used on the other half of the bed.

The data processing for the sleep phase recognition is based on detecting the sensor data variations in order to analyze movement. Discriminating between sleep phases using movement is a common approach that has been used in the past [SL86]. Using a sparse set of sensors it is possible to detect movement by comparing subsequent sensor readings and associate it to different sleep phases using different activity profiles. The system is based on the same prototype as the posture recognition system [DBM14].

4.3.2.1. Evaluation

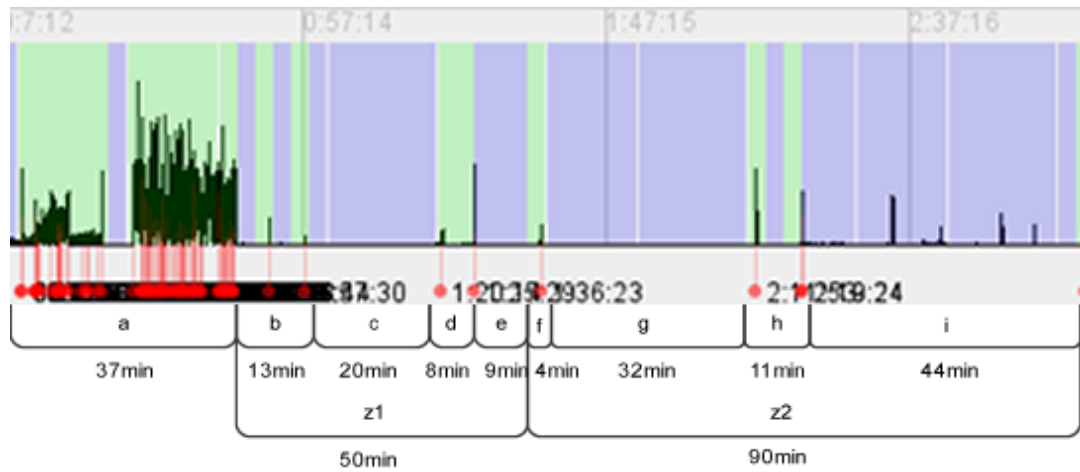


Figure 4.29.: Sleep movement data over three hours in one night [DBM14]

The Smart Bed posture recognition is able to successfully distinguish eight typical sitting and lying states. Using adaptation of the intermediate values it is possible to fit the state to an actual position on the bed, e.g. a *person sitting on the right side of the bed* state can be modified to any location on that specific side of the bed.

Regarding the detection of sleep phases there has been an evaluation and benchmarking of three nights [DBM14]. The Smart Bed was able to achieve a comparable performance to smartphone applications that detect sleep phases based on accelerometers. Figure 4.29 gives an example of movement recordings using the capacitive proximity sensors over one night. The activities are grouped into distinct chunks that are later associated to the sleep phases. Currently, breathing rate detection is added to the Smart Bed that can be used to improve the sleep phase detection and also can potentially detect anomalies that may be indicative of a certain health risk.

4.3.3. The Capacitive Chair

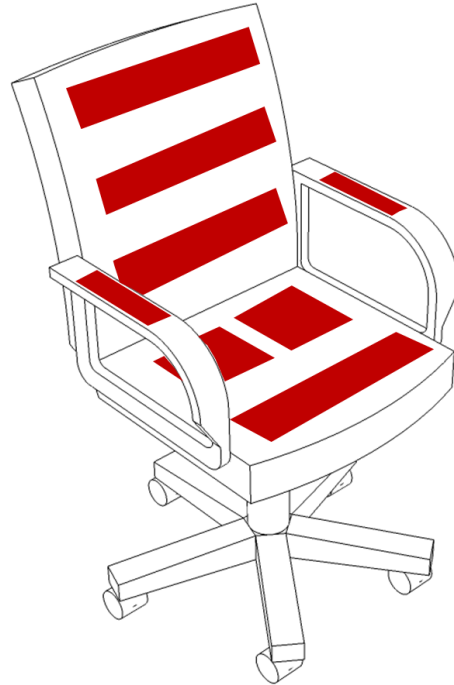


Figure 4.30.: Smart office chair sketch - eight electrodes three in backrest, three on seat and two in armrests

The Capacitive Chair is a regular office chair equipped with eight capacitive proximity sensors that can detect different sitting postures and work-related stress levels by examining movement and breathing rate, developed in collaboration with Sebastian Frank [BF14]. Seven solid copper electrodes that are placed below the covering are augmented by a single conductive thread electrode that is placed in a mesh on the backrest. In the past, smart chairs have used pressure sensors to infer posture and occupation [TSPM01]. Combining presence and proximity sensing, it is possible to directly infer postures where parts of the body do not touch the surface. Examples include the body being arched towards the front, or an arm being raised from the armrests. Additionally, higher area electrodes in the backrest allow detecting the breathing rate by measuring the movement of the chest.

The Capacitive Chair aims at providing different services to typical office workers and office managers. Using the occupation detection, it is possible to advise for some type of physical activity, if the time spent in front of the screen was too long. The system can also advise the user to change to a more back-friendly posture, or regularly switch the stance to achieve a more balanced load on the back. Using the breathing rate detection, we are able to get some sort of measure of the current stress level associated to a working situation. By adapting the environment, it is possible to improve the working atmosphere and reduce stress. The Capacitive Chair uses a multifaceted data processing approach. A machine learning algorithm is associating the sensing data to one of nine different typical sitting positions, inspired by a recent study of sitting positions for modern device usage [Ste13]. An adaptive body model that is fitted to the current sensor values allows for fine grained adaptation

of those postures. Finally, a combination of Fourier and data variation analysis is calculating the current breathing rate [BF14].

4.3.3.1. Capacitive layout

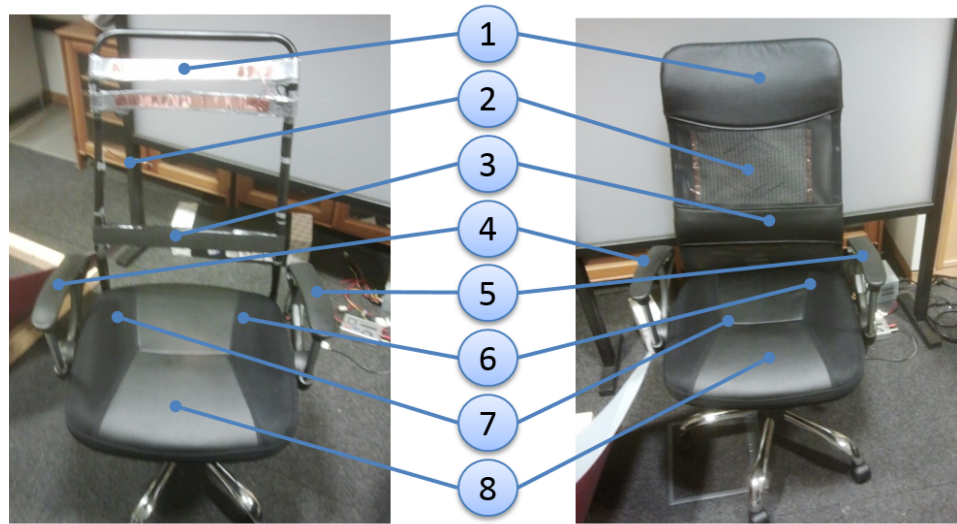


Figure 4.31.: Capacitive Chair electrode positions

The Capacitive Chair is based on a single OpenCapSense board that supports eight different electrodes. In order to get suitable posture measurements, the electrodes have to be equally distributed on the different areas of the seat. The measurement of the breathing rate requires a larger electrode near the chest area. Consequently, the electrodes are placed as follows:

1. Electrode on the upper part of the backrest (covered by faux leather)
2. Electrode in the central part of the backrest (using conductive thread)
3. Electrode in the lower part of the backrest (covered by faux leather)
4. Electrode below the right armrest
5. Electrode below the left armrest
6. Electrode for the left hip area below the left part of the seat
7. Electrode for the right hip area below the right part of the seat
8. Electrode for detecting both legs below the front part of the seat

The electrode in the central part of the backrest is based on conductive thread that was woven through the mesh material of the Capacitive Chair. The ends of the thread are connected to each other using conductive copper foil, in order to avoid negative effects that can occur when a single long electrode is used. A detail view is provided in Figure 4.32.



Figure 4.32.: Detail view of conductive thread electrode

4.3.3.2. Processing

The first steps of data processing are filtering using an 8-sample median filter, and a baseline calibration for all channels. Additionally, there is a dynamic normalization based on updated minimum and maximum values. For further processing, information is needed about the short-term value variance. This is gathered by calculating the difference quotient using a sample of ten average filtered measurements. Afterwards, a FFT is performed every 40ms and discretized into 512 values. A binning operation is performed to look for significant signals in a reasonable frequency interval (0.1Hz-3Hz) to detect the most significant frequency associated to the breathing rate.

The method to calculate poses has been described in detail in Section 4.2.2. The processed values of all sensors are compared to previously trained sitting positions of a user. The result of this process in the created demonstration software is shown in Figure 4.33. The fine skeleton model output is shown on the left, the static gesture classification probabilities on the top right and a visualization of the resulting posture on the bottom right.

Figure 4.34 shows a screenshot of the demonstration software currently performing the breathing rate detection. On the right side are the results of the FFT detection, showing the part of the signal relevant and the translation of the frequency to respirations per minute. On the left, there is the curve of sensor values showing the sinusoidal characteristics of the response to chest movement.

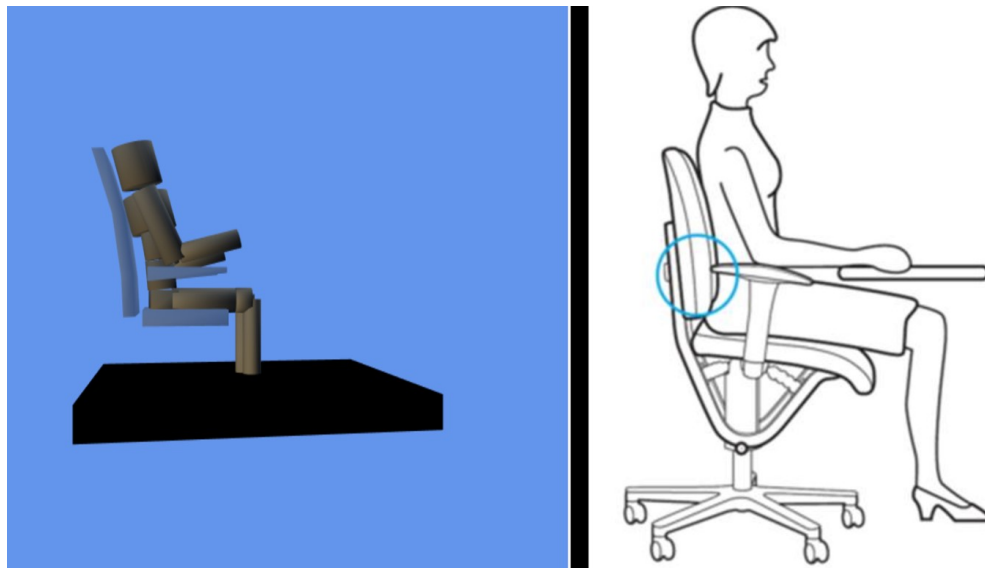


Figure 4.33.: Screenshot of the Capacitive Chair application showing the fitted 3D model on the left and detected posture on the right

Distinguish work activity levels

An additional feature of the software developed for an EIT ICT Labs project, called Cognitive Endurance during 2013, is a recognition module for working activities throughout a day. The method works similar to the sleep phase recognition described in Section 4.2.5. Aggregated sensor values are grouped into three potential activities, based on their variance over a certain amount of time.

- Active work as indicated by a certain level of movement while on the chair, e.g. gesturing while doing a phone call or performing small exercises
- Passive work as being present on the chair while not moving a lot, e.g. reading a web page or writing a document
- Not present at desk, whereas no one is currently sitting on the chair.

This results in an activity graph as shown in Figure 4.35 that creates an activity profile over the working day and can augment other activity tracking systems or act as self-standing system that gives employers and employees feedback on the type and quality of work.

4.3.3.3. Evaluation

The Capacitive Chair was evaluated in two distinct studies. The first aimed at testing the aggregated recognition of working activities with several persons over various days. The second study was testing the posture recognition with various users that were additionally queried about their general impression of the system. In this section we are presenting results of both studies.

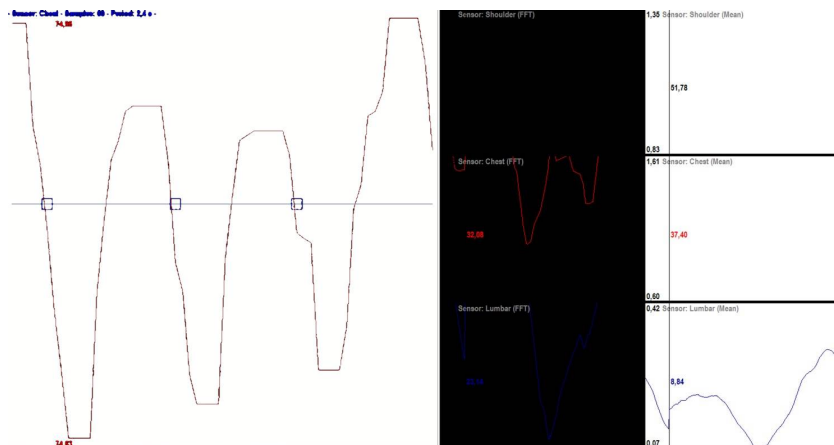


Figure 4.34.: Screenshot of the Capacitive Chair application showing the sensor values on the left and FFT analysis of various sensors on the right

Working situation recognition

The sensing chair supports distinguishing two different working situations that are determined using the method described in the previous section. The system also supports sending to the Cognitive Endurance server.

Figure 4.36 shows an example of this generated activity log. We have performed a test over three days between December 4th 2013 and December 6th 2013 on a typical work day in the office. The resulting activity logs were used to generate a chart as shown in the previous section.

The phases of not at chair can be seen clearly. Usually, they are attributed to lunch break or some meetings. The work is distributed between active work, such as writing and typing and longer phases of inactivity (such as reading). More results can be found in Appendix A.1.

Posture recognition

In a second evaluation the posture recognition of the chair was tested in a short study with 10 participants. Our system was tuned to distinguish three poses and a non-pose:

- Sitting upright
- Sitting hunched
- “Slouching on chair”
- Close to chair - disturber

The persons were given a short introduction, the different postures were displayed, and finally the persons were asked to perform the postures in order. When testing “close-to-chair” the subjects were asked to rattle at the chair, stand close, move it around and thus disturb the potential sensor readings. Each class was tested for 10 seconds, collecting 200 samples. Overall the results were very convincing. Of the 40 different measurements series only two were not achieving 100% accuracy. The Upright and Disturbance positions were classified correctly for all candidates. A single candidate had an 86% rating on the hunched posture. A different candidate had a 55% rating on the slouching position. The average of correctly classified postures is 98.5%. Detailed results can be found in Appendix A.1.

4. Use cases for capacitive proximity sensors

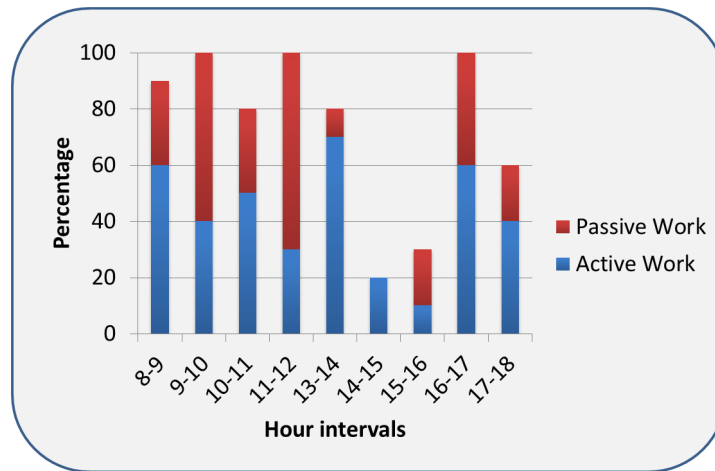


Figure 4.35.: Work Activity aggregation over a single work day (mock-up)

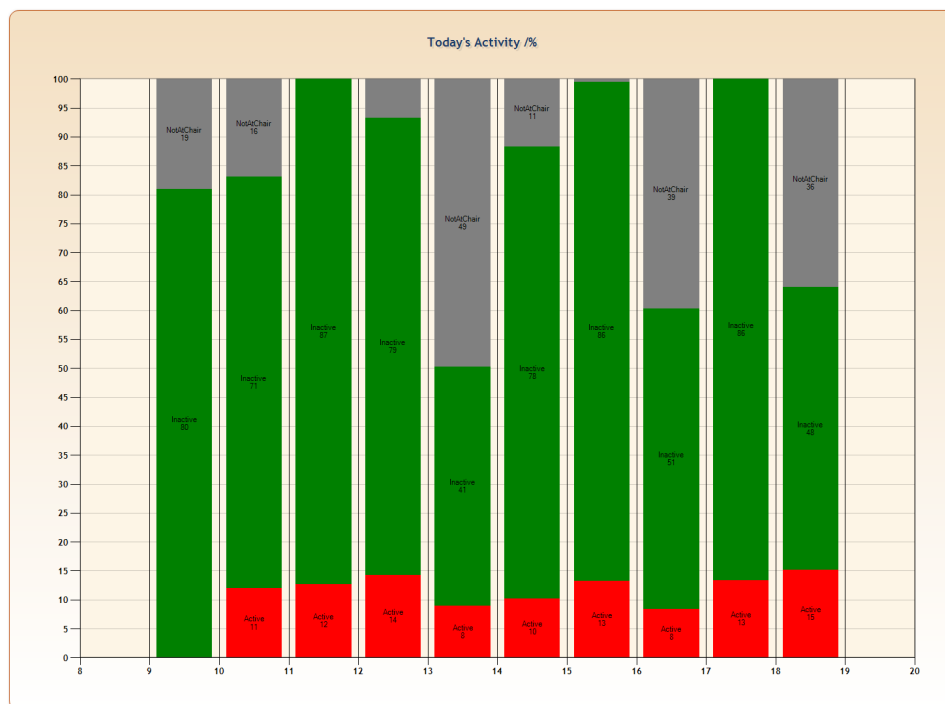


Figure 4.36.: Example chart of work activity data collected

4.3.4. Active Armrest

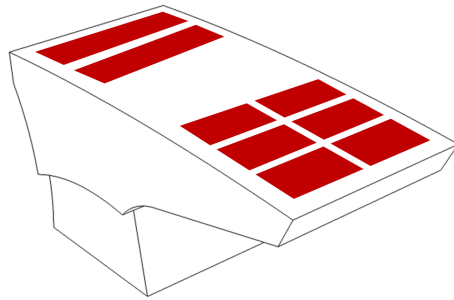


Figure 4.37.: Active armrest sketch - six electrodes for finger gesture detection in front, two for arm detection in back

Touch screens are by now also a trend in vehicles, with touch screens and touch pads becoming more common. The Tesla Model S provides a large area touch screen that completely replaces conventional button-based interfaces. However, touchscreens have been identified as potentially distracting for the driver [RB13]. My idea was to create a gesture input device based on capacitive proximity sensors, and unobtrusively integrate it into the car interior [BNS*14]. A suitable area for creating an interactive zone is the armrest, as it is the intended resting position in the first place. However, this creates an additional challenge. As the majority of interactions between arm and armrest are not intended to control aspects of the car system, we need concepts to infer the intention of the driver to interact with the car. The method has been described previously in Section 4.2.3. A sketch of this concept is shown in Figure 4.37. To test the validity of the invisible interactive areas and the two interaction concepts, we have created the Active Armrest, a prototype comprised of an aftermarket armrest with an integrated heterogeneous array of eight capacitive proximity sensors - two using large electrodes for detecting arm proximity and orientation and six using small electrodes to create an interaction area in front of the armrest. The system uses the classification method described in Section 4.2.3.



Figure 4.38.: Active Armrest prototype, left - outside view, right - detail view of electronics [BNS*14].

In order to evaluate the Active Armrest we have built the prototype shown in Figure 4.38. An aftermarket armrest was equipped with an OpenCapSense toolkit. The kit had to be modified due to the constrained interior and uses fixed wiring instead of USB connections. The demonstration application is based on the SenseKit debug software supplied with the toolkit. As of now there is a simple USB connection to a nearby PC. The gesture recognition framework was implemented in Java using the WEKA machine learning framework for SVM classification. A car interior demonstration application was created using Java and the Swing framework and mimics typical menu systems found on a touch screen.



Figure 4.39.: Postures of limbs on armrest - resting position (left), arm raised position (middle), hand raised position (right) [BNS*14].

In Figure 4.39 we can see the three different positions arm and hand can have on the armrest. On the left, arm and hand are in resting position with both close to the surface. The middle image shows the arm raised position and fingers touching the front of the armrest. The right picture shows the arm resting on the back and the hand in proximity of the front area. The system is able to distinguish between the three different positions using the methods presented in Section 4.2.3. A set of four different gestures has been defined for both interaction methods. The number is sufficient to control the user interface that was developed and supports both navigation and selection. The type of gestures has been defined after looking at previous research into touch and hand gestures [BNLH11, WKSE11]. For the touch interaction, left and right swipes performed with either one or multiple fingers are supported. Regarding the free-air interaction we are using left and right swipes, as well as planar circles either clockwise or counter-clockwise.



Figure 4.40.: Active Armrest demo software, left - finger tracker, right - OSM based navigation application [BNS*14]

Figure 4.40 shows two screenshots of the created demonstration application. It allows to control radio stations, selecting different audio files and looking at images. It is controlled using the gesture sets explained above. The

applications use a Next/Previous pattern for navigation within a UI level and Select/Back gestures to switch between the different levels of the UI.

4.3.4.1. Evaluation

	SSR	SSL	MSR	MSL		SR	SL	CLC	COC		
SSR	40		4		SR	36	6		2	SSR	Single Swipe Right
SSL	3	34		7	SL	8	35		1	SSL	Single Swipe Left
MSR	6	1	35	2	CLC	7	8	24	5	MSR	Multi Swipe Right
MSL		5	1	38	COC	5	18	1	20	MSL	Multi Swipe Left
										SR	Swipe Right
										SL	Swipe Left
										CLC	Clockwise Circle
										COC	Counter-clockwise Circle

Figure 4.41.: Confusion matrices of recognized gestures for touch interaction (left) and free-air interaction (right)

We performed a study with 11 participants investigating three different aspects - the detection rate of the gesture recognition system, differences in interaction speed between the two methods and getting a general feedback on the usability of our system. After a short introduction, the participants were asked to perform each gesture 4 times for both sets in alternating starting order. The results are shown in Figure 4.41. The touch interaction performed reasonably with a detection rate between 79.5% and 90.9% for each gesture. The detection rates of the free-air interaction were considerably lower, ranging from 45.5% for counter-clockwise circles to 81.8% for swipes to the right. The main issue is distinguishing between single and multi-touches. A personalized threshold that is calibrated to the user might alleviate this issue in future iterations. The interaction zone above the finger area is limited to a range of about 15cm. While performing the circular gestures the participants often left this area, leading to misattribution to swipe gestures.

In the second part of the evaluation the participants had to perform a task in the presented demonstration application - selecting and playing back a certain music file. We calculated the time required to perform the task. The average time for free-air gestures ($\mu = 125.67s, \sigma = 95.12s$) was considerably higher than the average task completion time for touch gestures ($\mu = 34.26s, \sigma = 28.61s$). It is noticeable that there is a very high deviation of the different runs, while the touch gestures fare better in general. While many users were able to quickly perform the task, others had a high number of errors and required several minutes. We can assume that a certain amount of training can reduce the required time. A trained user not participating in the study required 11s for the touch gestures and 18 s for the free air gestures.

Finally, we asked the participants to fill a general questionnaire on their experience with the Active Armrest comprised of a number of Likert-scale (1-10) questions. There was a strong preference for the touch gestures, in line with the results of the interaction time and gesture recognition study (1=touch gestures, $\mu = 1.72, \sigma = 0.84$). Most participants could imagine using the system for a longer period of time in their cars (10=strong agree, $\mu = 8.00, \sigma = 2.00$) and considered the device intuitive to use (10=strong agree, $\mu = 8.36, \sigma = 1.67$) and is an interesting interaction device (10=strong agree, $\mu = 8.63, \sigma = 1.26$). The touch interaction pattern was considered easier to use (10=very easy, $\mu = 8.00, \sigma = 2.45$) than the free-air interaction (10=very easy, $\mu = 3.64, \sigma = 2.68$). The opinions on the device precision were mixed (10=very precise $\mu = 7.27, \sigma = 2.43$).

4.3.5. Magic Box

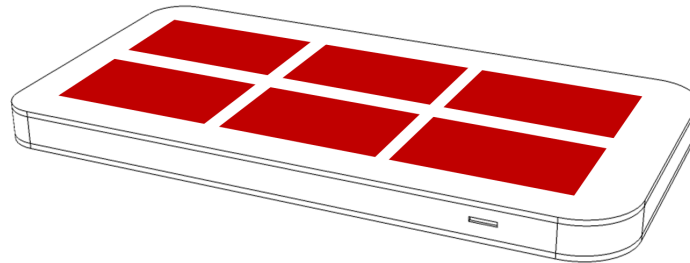


Figure 4.42.: MagicBox sketch - six electrodes uniformly distributed below surface

The so-called MagicBox was my first attempt to create an interaction device based on capacitive proximity sensing. For this prototype I collaborated with colleagues Pascal Hamisu, Tim Dutz and Felix Kamieth, as well as student Sebastian Jeckel. It is using an array of six individual wireless capacitive sensors that communicate to a central station [BH11]. The system was later extended to support Machine Learning based gesture recognition adapted from mouse gestures [BDK13]. The electrodes are using a large surface area and are made of aluminum foil. A sketch is shown in Figure 4.42. The system is able to track the position of a single hand in three dimensions up to a distance of approximately *20cm*. It uses different methods to infer gestures from the hand movement. It is designed to be a generic interaction device that can potentially be hidden below non-conductive surfaces. As it can be used without touching, it is also applicable in sterile environments. A suite of demonstration applications has been created that showcase typical scenarios for the MagicBox. This includes multimedia applications, like image viewer and media player but also a 3D object viewer intended as demonstrator for potential medical applications, allowing a surgeon to check MRT or CT images in a sterile environment without touching any surface.

4.3.5.1. Evaluation

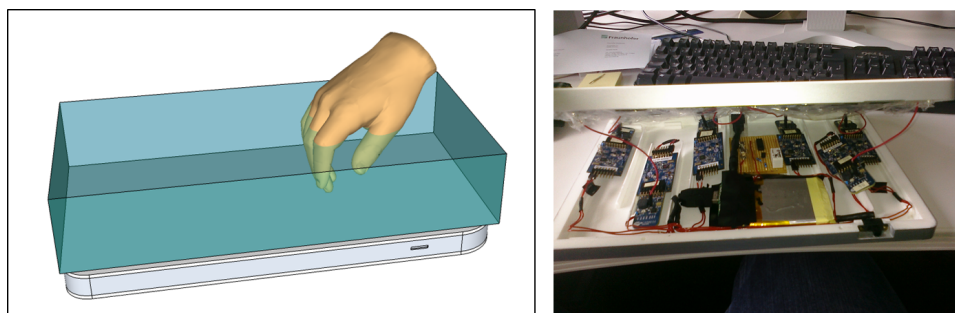


Figure 4.43.: MagicBox conceptual rendering (left) and detail view of electronics (right) [BH11]

The MagicBox prototype is based on the Cypress First Touch starter kit [CC13] and combines six capacitive sensors communicating wirelessly to a single base station. They are put into a casing together with an USB-rechargeable power supply. A conceptual rendering showing the interaction area and a detail view of the prototype electronics are shown in Figure 4.43.

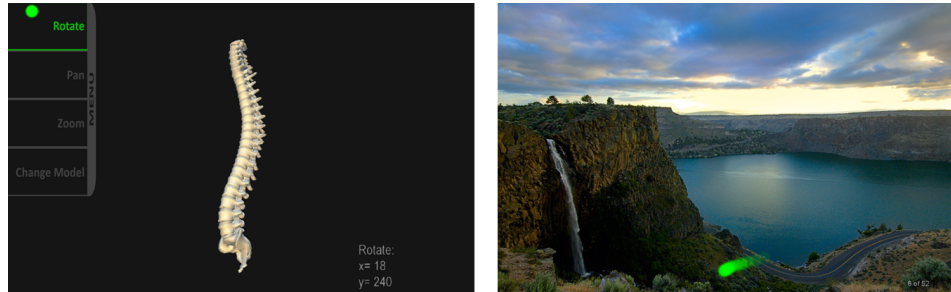


Figure 4.44.: MagicBox demonstration application - 3D object viewer (left) and image viewer (right) [BH11]

The different iterations of the MagicBox have been evaluated in conjunction with various demonstration applications. A usability study with 18 persons led to general approval of the system [BH11]. Two of the applications used in this study are shown in Figure 4.44. On the left, there is a 3D object viewer that has to be controlled by a combination of menu and direct manipulation of the screen content. On the right side there is an image viewer that was controlled by gesture to trigger the next/previous images or perform zooming operations. The most common positive remarks gathered in this study can be roughly put into three groups:

- The device very intuitive to use
- The idea of interacting this way is novel and interesting
- It is easy to control applications with those gestures

Likewise we identified three main groups for negative comments about the prototype:

- The device is not very precise
- The interaction speed is slow
- It can be tiring for the arm

Later iterations have been trying to improve some of the weaknesses presented above, e.g. by using a more sophisticated gesture recognition system and faster sensor refresh rates. Accordingly, there were fewer complaints about interaction speed and precision [BDK13]. However, the final complaint about the device being tiring for the arm, requires a different approach, that we are investigating in the next prototype to be presented in this section.

The overall method is similar to mouse gesture recognition, albeit adapted for three dimensional locations. The developed system allows defining an arbitrary set of potential gestures and adding training data to each, as explained in Section 4.2.1. The key aspect of gestures by example – providing examples - is realized in a debug application. It provides a simple way to record exemplary movements and associate them to gesture sets. The main screens realizing this functionality are shown in Figure 4.45. On the left side we can see the management screen that allows adding and deleting gestures, as well as a preview that is an average of the sample data associated to this gesture. The process of entering data is shown on the right side, where several samples can be recorded and associated to the selected gesture. The user then can decide, whether the current movement should be stored or discarded.

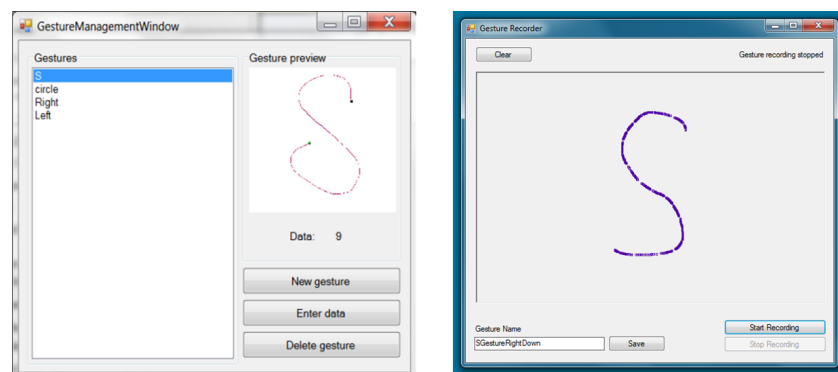


Figure 4.45.: Gesture overview module (left) and gesture recorder (right) [BDK13]

4.3.6. CapTap

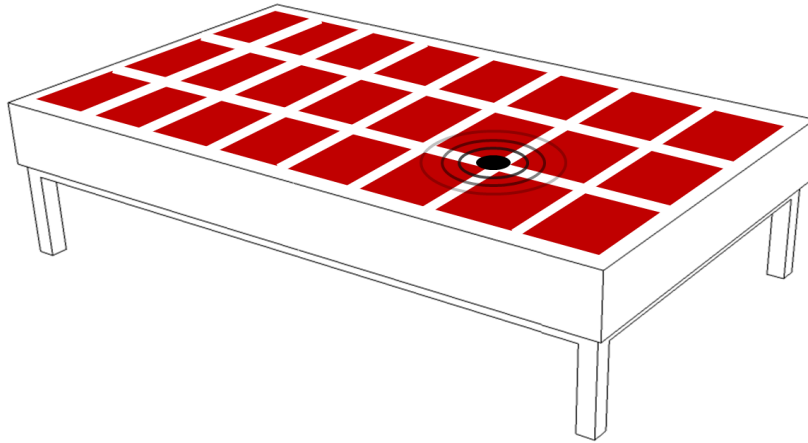


Figure 4.46.: CapTap sketch - 24 electrodes placed under table surface and a single detector for touch events

A general insight of free-air gestural interaction that became apparent even in early works is the physical demands of prolonged interaction with such systems [BBL93, LBT02] and the difficulty to adapt selection events to gestural input - the latter typically being realized by time- or position-based gestures [BBL93, KOR*02]. There is no trivial solution to this challenge and any approach has to take into account the specific application scenario covered. Some systems attempt to provide specifically adapted graphical interfaces, while others include additional input devices assisting the interaction [WB03, ZLB*86]. A major point is decreasing the required time for interaction, e.g. by adding a tactile feedback to the interaction system, preventing time-based selection gestures. CapTap is a regular living room table that includes a capacitive proximity sensor array for tracking the position of one or more arms. As it is difficult for this sensor category to detect touch, if the interaction surface is at a distance from the electrodes, a hidden acoustic system is added that allows recognizing different touch events. CapTap tracks hand and arm position using the image-based object recognition previously presented and fuses this data with different touch events generated by a single contact microphone that analyzes the audio signals in frequency space. This allows to significantly reduce interaction time, as opposed to systems relying on time-based dwell gestures. The system was created in 2013 and 2014 with collaboration of several students, most notably Sebastian Zander-Walz and Stefan Frank [BZWK*14]. It is used and further developed within the European research project POSEIDON that aims at providing technical solutions to help persons with Down's syndrome on planning their day. The idea is to have an input device that allows interaction regardless of motoric skill level.

4.3.6.1. Capacitive layout

The goal of CapTap is to enable tracking of hands and arms that move above a table. Therefore, the electrodes are placed in a uniform array that provides similar sensing properties for the whole surface. It is realized as a prototype installed in a regular living room table. It is comprised of an array of capacitive proximity sensors, a contact microphone for touch event detection and a miniature PC. All devices can be integrated into the table in a way that it is not distinguishable from the not-augmented piece of furniture. Figure 4.47 shows some detail

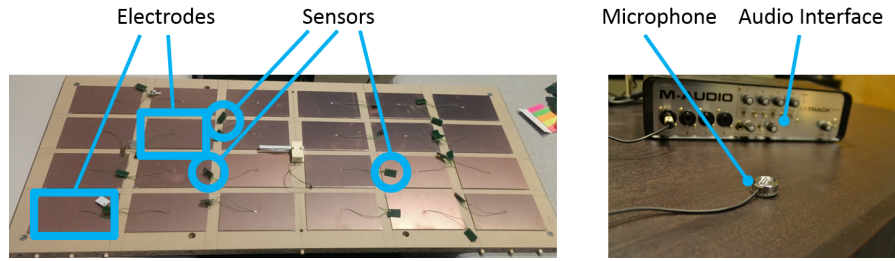


Figure 4.47.: Detail views of the prototype system: left - electrodes and sensors, right - audio interface and contact microphone

views of the disassembled prototype. The left image shows the back of the wooden tabletop. The electrodes are arranged in a 6x4 array and each one is attached to a single sensor. The right picture shows the touch detection microphone. It is attached in the center of the surface, as to avoid non-uniform sound distribution over the surface area that would be more difficult to train. Placing the microphone below the surface has no strong influence. However, a specific training phase is required for any novel surface that is equipped with the touch detection devices.

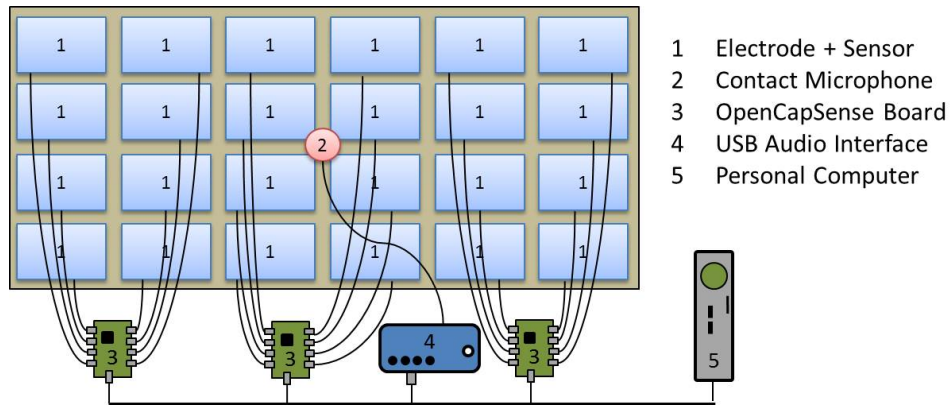


Figure 4.48.: Abstract view below the surface of the prototype including capacitive sensing electrodes and touch detection microphone

The setup is visualized in Figure 4.48. The system is comprised of 24 electrode & sensor pairs that are connected to three different measurement boards. The microphone is attached to a USB Audio Interface. Overall there are four different boards connected via USB to a PC that executes and merges the different types of data processing and links it to the software suite. The prototype is based on OpenCapSense, an advanced prototyping system presented by Grosse-Puppendahl et al. [GPBB*13]. The boards are performing some prefiltering, whereas the image-based hand tracking is realized on an attached PC. The microphone is attached to an USB audio interface (M-Audio M-Track Quad) that transfers data acquired by up to four microphones and provides various pre-sampling functions. All four devices are attached to a Mini-PC that is performing subsequent data processing and is running the demonstration and testing applications.

The final table prototype can be seen in Figure 4.49. On the left side the table is seen without any additional markers - on the right side we see the table equipped for the evaluation using a set of markers for different touch



Figure 4.49.: Views of final prototype, complete view (left), top view with markers for touch evaluation (right)

and swipe events. The debug software was developed with C# using the .NET 4.5 framework. We are using the Emgu CV library based on OpenCV for image processing and application of the Kalman filter to the determined palm locations. The sound processing is implemented in C++ and Java using a modified version of Chuck for audio sampling and the WEKA framework to apply the machine learning on top. We are using sockets to transfer data between the different modules. The debug application allows a fine control of the various processing steps in both image and audio signal processing.

4.3.6.2. Applications

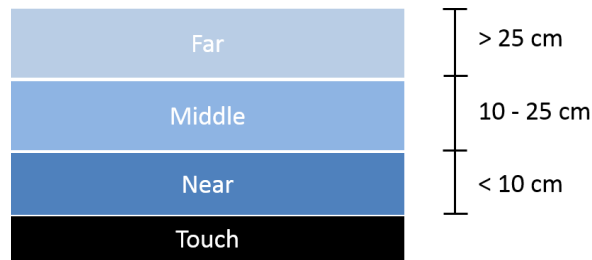


Figure 4.50.: Multiple interaction layers

The tracking capabilities of CapTap enable a variety of different interactions. In this section we present three demonstrations that are organized based on the following novel aspects:

- Depth sensing and layer based interaction
- Using arm orientation and position as input
- Impact and swipe touch events

Depth sensing and layer-based interaction There are several limitations of free-air gesture interaction systems, related to user fatigue [BBL93, LBT02]. Additionally, the achievable resolution above the surface is considerably lower than on the plane. If these two aspects are combined, an interaction pattern that includes a tactile sensation for improving selection events in graphical user interfaces and a sparse quantization of the interaction space above the surface seems viable. Thus we are proposing a layer-based interaction for capacitive proximity sensor devices that is comprised of a touch layer that can be used to register several different types of touch events and three different distance layers based on the proximity of the hand [BZWK*14]. This is inspired by similar multi-layer patterns introduced by Subramanian et al. for pen-supporting tabletop interaction systems [SAL06].

The different layers in Figure 4.50 are intended for touch and free-air interaction. The touch layer at the bottom of the Figure represents the interaction on the surface of the table. The layers on top are formed by dividing the area above the table in three parts – the near layer, the middle layer and the far layer, realized using the elevation calculation presented in the sparsely distributed algorithm section. Inside each layer a set of different gestures can be executed and recognized while inter-layer changes may trigger additional events. Different interaction types are used for the touch and the free-air gesture layers. Swiping and dwelling gestures are suited for the free-air gesture layers while at the touch layer tapping, double tapping and different swiping patterns can be used. These interaction types may have a different functionality at each layer.

One potential application is the rapid access to large data sets, e.g. images. We developed an image viewer that enables access to large collections of pictures. In the far layer, it is possible to quickly scroll through hundreds of thumbnails. If the hand enters the middle layer a more detailed scrollable view showing fewer thumbnails is triggered. In the near layer the user can select an image using a cursor. If the table is touched the selected image will be shown in full screen. The different sizes of far layer and middle layer are shown in Figure 4.51 on the top.

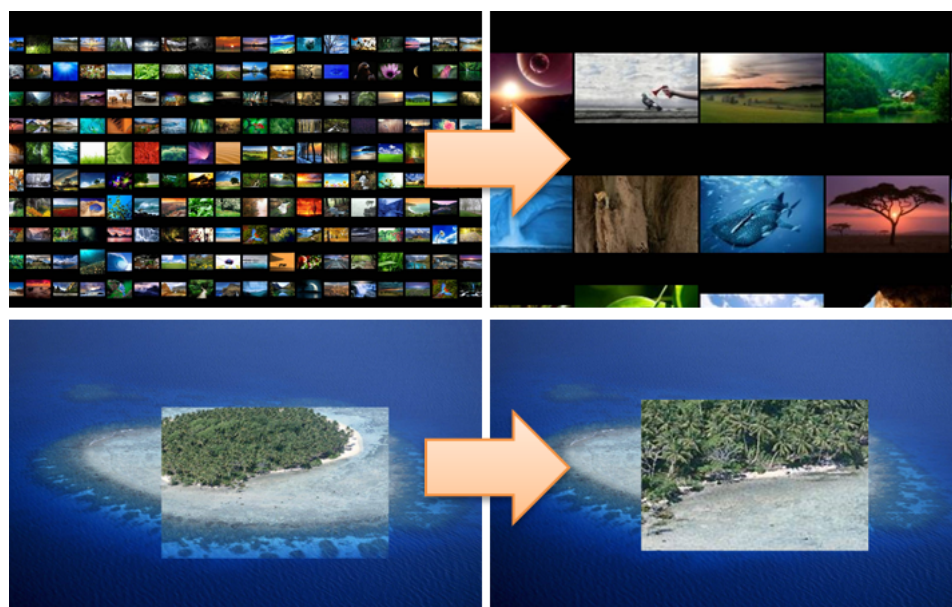


Figure 4.51.: *Top*: Progression from far layer to middle layer image thumbnail sizes. *Bottom*: Different zoom levels controlled by arm orientation.

Input from arm orientation and position The knowledge of arm position and orientation can be used to enrich the interaction. A first example is the differentiation of users on the Medusa tabletop [AGWF11]. Knowing the arm position and orientation, we can identify from which direction the arm is approaching the table.

In a single-user environment we can disallow interaction from arms entering from a side that is not associated to the primary user. In multi-user environments personalized content can be delivered. An example is shown in Figure 4.52 on the left. Additionally, the orientation can be used directly as additional input parameter. In a demonstration application we adjust the zooming factor of the screen lens, as shown in Figure 4.51 on the bottom.

If the arm is held straight the factor is low and increases with the rotation angle. This requires the person leaning closer to the screen and thus mimics the natural reaction when something is difficult to read. This is shown in Figure 4.52 on the right.



Figure 4.52.: *Left*: Recognized disturbing arm. *Right*: Arm orientation when leaning in.

Impact and swipe touch events The different categories of touch events can be used to enable a variety of different interactions. Another simple demonstration application is the painter application. The touch layer is used for drawing. Using taps the user can cycle through different colors. With knocks the brush width can be adjusted (single increase, double decrease). Additionally, with a full hand swipe the whole canvas can be erased.

4.3.6.3. Evaluation

To evaluate CapTap we have performed a variety of different tests. The first is a test of the hand tracking, followed by two user studies. There was a combined number of 23 participants. In both studies we included questionnaires to capture the experiences of the participants. All non-open questions used a Likert scale from 1 to 10 that we will refer to with average and standard deviation. The first study included 13 participants that were invited to test the accuracy of the touch detection. This group had a high experience with touch devices ($\mu = 9.00, \sigma = 1.96$). The second study with 10 users was intended to benchmark the interaction speed and get a feedback on the usability of the different demonstration applications. This second group predominantly had plenty of experience with touch devices ($\mu = 9.40, \sigma = 1.90$). Experience with gesture interaction systems, such as Kinect or Leap Motion, was less prevalent and had a higher variation ($\mu = 6.00, \sigma = 2.71$).

Hand tracking Tracking the accuracy of capacitive free-air systems can be challenging, as the electrical properties may vary over time and according to object shape. Thus, typically only static conductive objects are evaluated instead of hands [GPBB*13, WKBS07]. The paths a hand performs is the most relevant measure, if any subsequent gesture recognition should be performed. A visualization module was added that shows the tracks followed by one or more hands and is added to an overlay of the image by the registered camera. Figure 4.531 shows several of the paths that were generated this way using tracking of a single hand that moves in the middle interaction layer. The system can create smooth trajectories that can be analyzed further. However, as the demonstration applications do not use classified gestures, but only palm position, it is not connected to any gesture recognition module.

Touch detection accuracy The main interesting aspect of this evaluation is to test if the classification method performs well, if the system is used by persons that did not participate in the training. Six different types of touch events are tested by the different users - finger tap (SFT), double finger tap (DFT), knuckle knock (SKK),

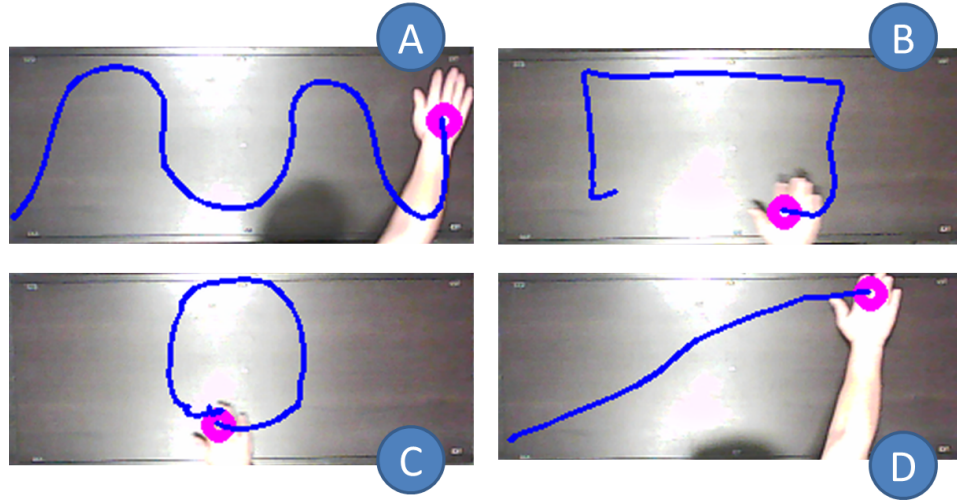


Figure 4.53.: Tracks generated by Kalman filtered palm position, a sine wave (A), a rectangle (B), a circle (C), and a diagonal swipe (D)

double knuckle knock (DKK), finger swipe (FS) and hand swipe (HS). In addition it is necessary to evaluate if outside influences can disturb the signal, thus the users stomp (STO) at three different locations. Overall there are 15 different areas on and around the table that have to be touched in different ways by the users. These are executed three times each, leading to 54 touch samples and 9 stomp samples per user. The locations shown in Figure 4.54 are (double) finger taps (1,2,3), (double) knuckle knocks (4,5,6), finger swipes (7,8,9), hand swipes (10,11,12) and stomps (13,14,15). The system was pre-trained and does not use training data from the participants. The subjects were shown all different supported touch gestures just once in a live example. The results of the evaluation are shown in Table 4.4. A very good classification for finger taps, knocks and finger swipes was achieved, ranging between 94% and 100%. Hand swipe detection was less precise, with a detection rate of just 84%. The environmental influence, as tested by the stomps around CapTap led to false-positive classifications in about 9% of all instances.

Table 4.4.: Results of touch detection for single and double taps (SFT, DFT), knocks (SKK, DKK), finger swipe (FS), hand swipe (HS) and stomp (STO). Noted are the overall samples, errors, no event errors, wrong classification errors and the percentage of correct classification.

	SFT	DFT	SKK	DKK	FS	HS	STO
Samples	117	117	117	117	117	117	117
Errors	7	1	1	1	0	19	10
No event	0	1	0	0	0	0	0
Wrong Class	7	0	1	1	0	19	10
Percentage correct	94.01	99.15	99.15	99.15	100.00	83.76	91.45

Two trends concerning the errors of the touch detection were observed. They are personal, as in attributed to just a small subset of the participants and related to direction for swipes. Six of the seven finger tap errors were

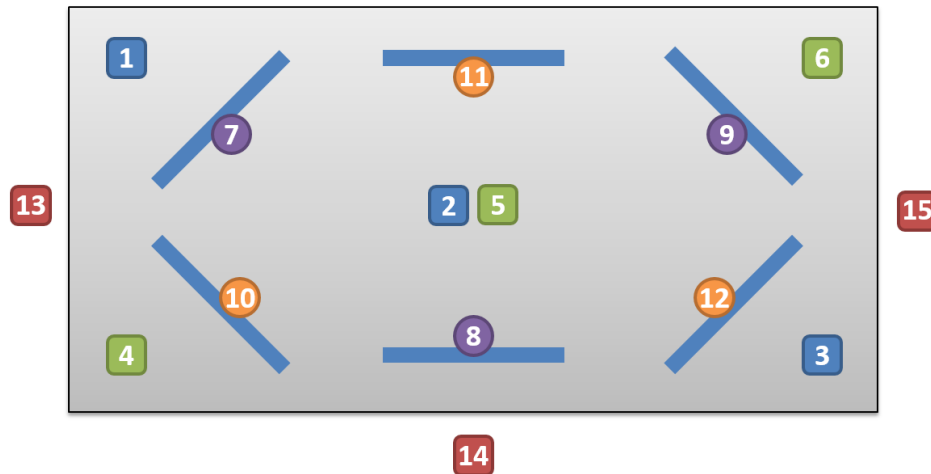


Figure 4.54.: Finger tap (blue), knuckle knock (green), finger swipe (purple), hand swipe (orange) and stomping (red) spots relative to tabletop.

produced by a single person. Similarly, for one person there was no successful registration of hand swipes, and two others had three or more errors. These participants accounted for about 84% of all hand swipe classification errors. Similarly, two persons produced 70% of all stomp misclassifications. This can be attributed to personal preference and variety in touch interaction. While the system worked well for most users (8 of 13 participants had an overall error rate of less than 5%), there should be some calibration routine that adapts the algorithm to the different users. The second observation was the directional dependency of swipe classification errors. Regarding the hand swipes, the diagonal swipes at positions 10 and 12 resulted in six errors each, while the horizontal swipe only resulted in three. The reason for this behavior is not entirely clear. It is assumed that the highly complex shape of the hand is causing different mechanical waves, depending on the swiping direction. This is not expected for finger swipes, as only the tip is involved and can be considered fairly uniform. This should be investigated further by considering classifying different swiping directions of the hand from audio alone. After the study the preferences of the participants were inquired. They preferred ($\mu = 3.62, \sigma = 2.36$) finger taps (Likert score 1) to knocks (score 10). There was also a clear preference for finger swipes (score 1) or hand swipes (score 10) ($\mu = 2.69, \sigma = 2.25$). This can be taken into consideration when designing interfaces with a constrained input set.

Interaction speed study A major point of interest for us was to check if users could successfully use the layer interaction pattern introduced before and if the option for adding unobtrusive touch detection would have any influence on the interaction speed. For this a small game was used, whereas the participants had to put a cursor into a box and perform either a dwell (approximately 300ms) or touch activity for selection. The cursor reflected the current interaction layer by color coding. There was a count of the time it took to complete a run of 15 boxes.

Some example boxes are shown in Figure 4.55. There were six different types referring to dwelling in the three different layers, knock, tap and disturber. Each participant performed four runs. First a training run, then a run with all interaction layers and the two touch events, then a run with dwelling boxes only (without layers), and finally a run with tap boxes only. The dwell and tap runs also included some disturber boxes to slightly increase the challenge. The order of the runs were switched equally between the different participants. Regarding the

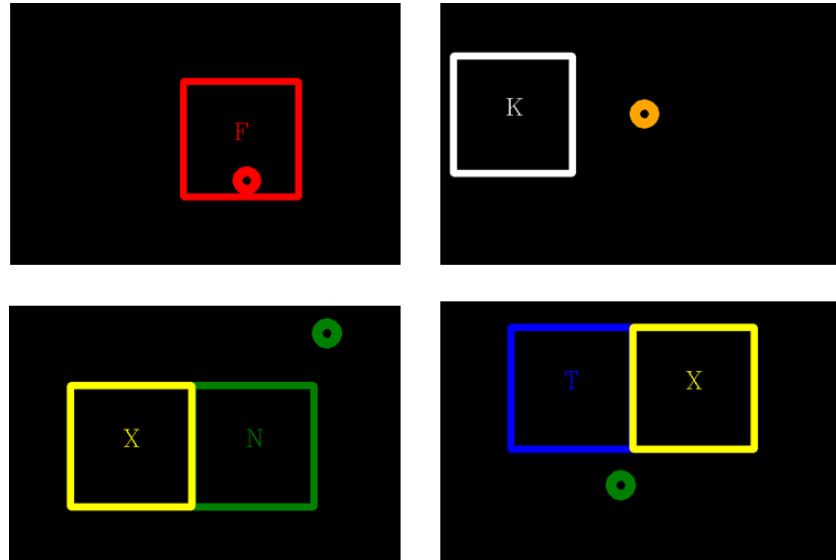


Figure 4.55.: Interaction speed evaluation. Different types of boxes for near layer (N), knock (K), far layer (F), and disturber (X).

full interaction run it is important to know if the user understands the different layers and can achieve a good interaction speed. The main hypothesis was that tapping improves the interaction speed as opposed to dwelling and may reduce the overall interaction times, thus reducing potential fatigue. It was expected that the tap run is shorter than the dwell run, as selection events can be performed faster. All runs had the same overall distance and the last two had inverse order to account for Fitts' law.

Table 4.5.: Results for interaction time in the different runs of the interaction speed test

	Full run	Dwell run	Tap run
Average Time	40.12	37.97	33.42
Shortest Run	27.28	32.20	24.29
Longest Run	51.38	48.11	47.72
Standard Deviation	7.22	5.27	7.93

The results are shown in Table 4.5. Running a paired t-test on dwell and tap run the resulting p value is 0.0071, indicating a high statistical significance that the interaction using taps is faster than the interaction using dwelling. This fits expectations from literature [LBT02]. While this can be countered by reducing the dwell time, the risk of wrongful selection of nearby objects increases significantly.

Experience questionnaire Finally, all participants were asked to fill in a questionnaire with Likert-style questions in the style mentioned at the beginning of this section. Most users considered the device to be easy and precise enough to use ($\mu = 8.1, \sigma = 1.52$) and considered it highly intuitive ($\mu = 8.9, \sigma = 0.74$). Regarding the layer model they had no problem using it in the full test run ($\mu = 8.6, \sigma = 0.97$) but were critical to adding more layers ($\mu = 3.9, \sigma = 2.08$). They clearly preferred ($\mu = 2.6, \sigma = 2.72$) finger taps (Likert score 1) to knocks (Likert

score 10). There was no clear preference for finger swipes (score 1) or hand swipes (score 10) ($\mu = 4.6, \sigma = 3.06$). The participants considered CapTap to be an interesting interaction device ($\mu = 9.1, \sigma = 1.20$) and could even imagine using it for longer periods ($\mu = 6.9, \sigma = 1.97$). Asking for particular points that they liked about the current version of CapTap the comments mentioned the ease of use and the high variety of different input commands that are supported. Points that were disliked are the usage of knocks that were considered unpleasant after a short while, even during the 10 minute study that only included few knocks. The hand tracking at this point is sometimes disturbed by the user's knees that enter the generated electric field around the table.

4.4. Other capacitive prototypes

In collaboration with other partners from industry and different students some additional prototypes have been created that are discussed briefly in this section.

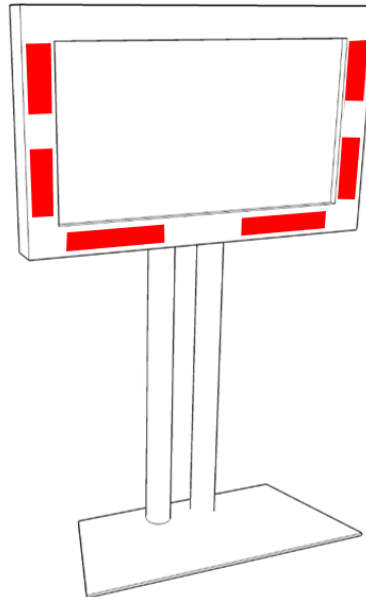


Figure 4.56.: CapDisp sketch - TV on a stand equipped with capacitive sensors hidden below a plastic cover

CapDisp is a presentation display augmented with capacitive sensors to enable touch-free control of several applications. It was created as a prototype for Hessen IT GmbH in 2010. The system is comprised of six Cypress CY3271 capacitive sensors that are powered via USB and interfaced to a Mini-PC that is attached behind a 42" display on a presentation stand, as shown in a sketch in Figure 4.56. The display is set in an additional case that hides the six electrodes that are made of copper foil. As shown in Figure 4.56, they are placed on the bottom and right part of the screen, allowing to detect four different swiping gestures, left and right on the bottom of the screen, up and down on the left side of the screen. The gestures can be performed at a distance of up to 20cm in front of the electrodes. Since the primary purpose of this device is showing presentations, some additional dwell gestures were added, that allow jumping to the first or to the last slide by holding the hand in front of a specific sensor for a certain time. Additionally, two other applications were included, a gesture-controlled image viewer and a video player.

The smart couch was created by then-student Tobias Grosse-Puppendahl in scope of a practical course in 2010/11, supervised by Alexander Marinc and me. The results were later published at the AmI 2011 conference [GPMB11]. Using an array of eight capacitive proximity sensors that are unobtrusively placed inside a couch it

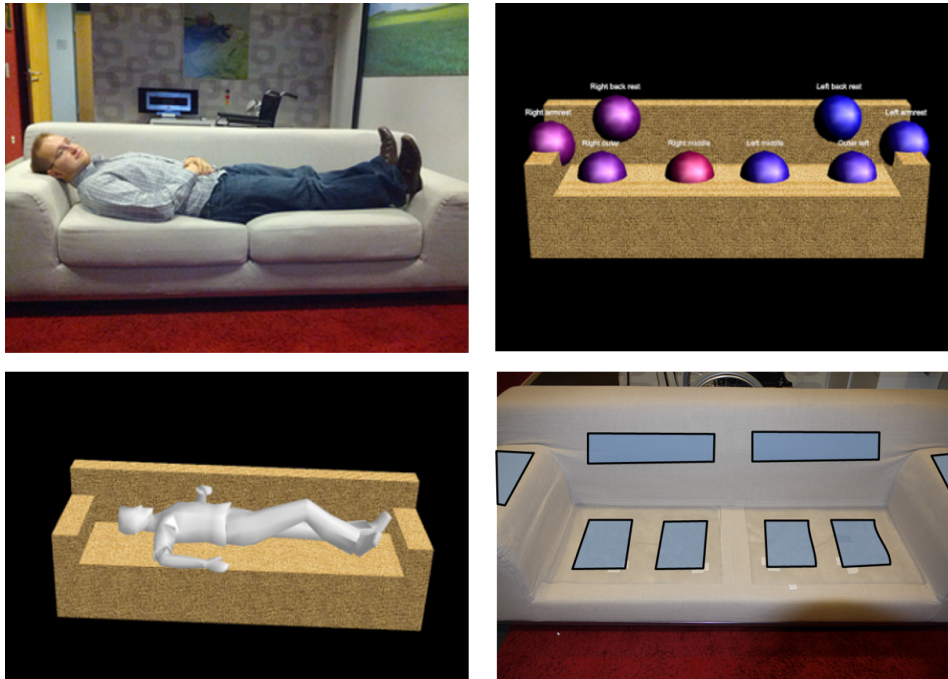


Figure 4.57.: *Top left:* Person lying on the couch. *Top right:* Resulting sensor value visualization. *Bottom left:* Rendering of recognized posture. *Bottom right:* Position of electrodes within the couch.

is possible to determine the posture of one or more persons that are currently occupying the system. The sensor readings are calibrated and normalized and fed into the WEKA machine learning framework for classification. Three different classifiers have been tested, decision trees, Naïve Bayes and RBF networks, whereas the latter provided the best results. The system was evaluated with 18 users and 8 resulting postures (6 with one person, 2 with two persons), such as sitting left or right, or lying in a specific direction. The resulting measurements was distinguished in a training set from 9 persons and a test set from 9 persons. The resulting precision was 97.5%, the recall 97.2%. The system is still working as a demonstrator in our living lab, implicitly controlling different networked systems based on the detected postures, e.g. activating ambient lighting as soon as the person is lying down.

Honeyfish is a gesture interaction device based on capacitive proximity sensors operated in shunt mode. It was created by Tobias Grosse-Puppendahl in scope of his Master's thesis that I supervised in 2012. It led to two different publications focusing on the provided contributions in multiplexing and object detection [GPB12] [GPBKK13]. The system is using eight different transmitters and two receivers, leading to a set of 16 virtual sensors that are set in the middle of each receiver-transmitter combination. As shown in Figure 4.58 on the top left, the receivers are in the center and the transmitters are placed on the outside. Using a frequency division multiplex it is possible to generate 50 samples from each virtual sensor. The system is using a method of object tracking that extends on a proposal by Smith, dubbed Swiss-cheese, which is based on the premise that each sensor not recognizing an object or a distant object is cutting a (ellipsoid) hole in the object presence probability space, thus leading to a probability distribution visually resembling a Swiss cheese [Smi96]. The remaining probability space can be analyzed to fit hand shaped objects, thus enabling gesture detection. A particle filter

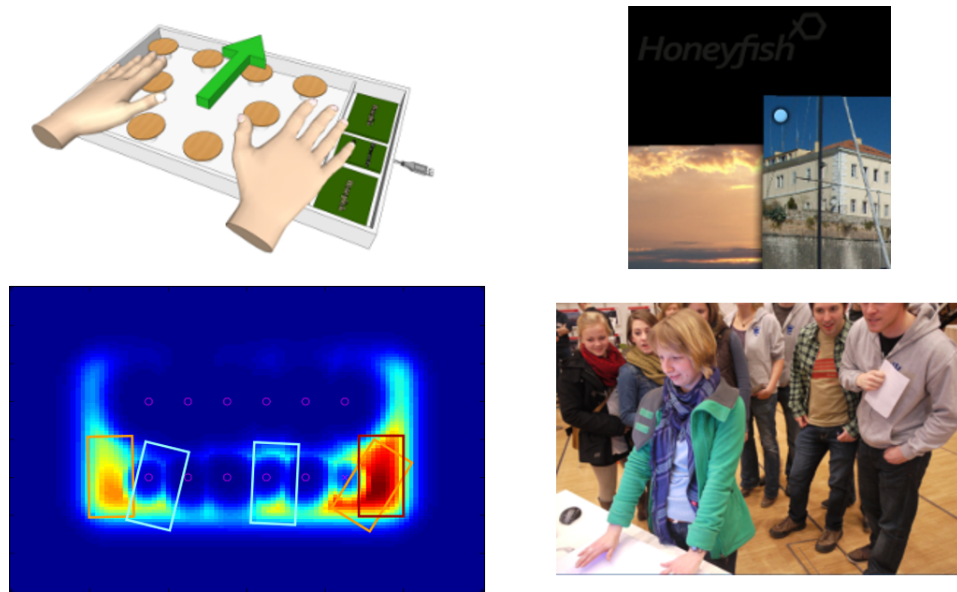


Figure 4.58.: *Top left:* Rendering of Honeyfish device and hands. *Top right:* Image of developed GUI and pointer. *Bottom left:* Swiss cheese algorithm predicting objects in interaction space. *Bottom right:* Evaluation of Honeyfish at a student fair.

is used to track the position of the hands. The system is able to track multiple hands and was used to control applications ranging from an image viewer, the GNU TuxRacer game to a remote-controlled vehicle.

GestDisp is the final prototype discussed in this work. It was created by Yannick Berghöfer in 2013 as part of his Master's thesis that was supervised by me and Tobias Grosse-Puppendahl. The basic idea is to enable gestures performed in front of a screen by applying capacitive proximity sensors on the screen surface. Consequently a different type of electrode material has to be used, that combines conductivity and transparency. Different materials have been evaluated, including ITO (indium-tin oxide) and PEDOT:PSS (a conductive polymer) in order to find a suitable electrode candidate. ITO was chosen and attached to the screen including shielding electrodes, reducing the effect of the electric components used to create this display. Nonetheless, the complex and highly disturbing environment drastically limits the distance in which gestures may be performed. The gestures are recognized using a Hidden-Markov-Model classifier that was trained by the developers. Using this, it was possible to distinguish swipe gestures in four directions, selection gestures as indicated by dwelling at a certain position and combined swipe-and-hold gestures, whereas the hand is resting in the interaction area after performing a swipe, e.g. used for continuous scrolling or increasing the volume. The gesture recognition includes a specific garbage-gesture that allows to distinguish arbitrary movements in the interaction area from deliberate gestures and has to be trained separately. This approach allows a precision and recall of about 95% on a collected training set. A demonstration application was created that mimics a multimedia system in a car, allowing to control different radio stations or select music of choice.



Figure 4.59.: *Top left*: Concept image of GestDisp in a car with outlined electrodes. *Top right*: GestDisp prototype installed in front of a monitor. *Bottom left*: Screenshot of demonstration application music player. *Bottom right*: Association of gestures to functions in the demonstration application.

4.5. Capacitive prototypes from related work

While I already have presented capacitive systems in the related work section, they are briefly revisited here, to classify them given the specified application domains. Similar to the created prototypes I will give some additional detail regarding their measuring layout and data processing. The prototypes are shortly listed in Table 4.6. This additional information will be used to give a more informed discussion of benefits and limitations of capacitive proximity sensing technology in smart environments and will contribute to the set of guidelines for developers of capacitive sensing applications.

Table 4.6.: Measuring layout and data processing of different prototypes from related works

Name	Description	Application Areas	Measuring Layout	Data Processing
SensFloor [LS09]	System for indoor localization and fall detection as floor underlay	Indoor Localization	Loading mode, variable number of sensors based on area size	Individual coding of zones on floor - analysis of activity based on trajectories
TileTrack [VMV09]	Indoor localization using transmitters below floor and receiver electrodes in wall or furniture	Indoor Localization	Transmit mode, large transmitter electrodes below floor, different receiving electrodes	Location by calculating center-of-gravity on most active tiles
Touché [SPH12]	Swept-frequency sensing to detect different types of touches on a conductive material	Smart Appliances	Swept-frequency sensing, single electrode	SVM classification using features in different frequency ranges
Botanicus Interactus [PSLS12]	Using plant tissue as conductive material as application for swept-frequency sensing	Smart Appliances	Swept-frequency sensing, electrode coupled to plant tissue	SVM classification of touches that are transferred to input events
HandSense [WB09]	Using capacitive sensors to detect grasp style	Smart Appliances	Four loading mode sensors on CapToolKit	Classification of holding styles based on threshold levels
Active capacitive sensing [CAL10]	Conductive textile electrodes to sense different parameters of the human body, based on location	Physiological Sensing	Loading mode, single electrode attached to body part	Different filtering methods, based on electrode position, activity classification using LDA
Spread spectrum sensor [Mac04]	Single electrodes using a spread spectrum technique for improved sensitivity	Physiological Sensing	Loading mode, single electrode placed remotely	Spread spectrum technique to improve SNR, amplitude measurement for respiratory rate
School of Fish [Smi99]	Array of shunt mode sensors that can track 3D position and orientation of two hands	Gesture Interaction	Shunt mode, flexible array of sensors	Modeling hands as collection of spheres and fit into area based on sensor values and position
Thracker [WHKS06]	Four electrodes placed around display that can sense spatial position of hand in front of display and certain gestures	Gesture Interaction	Loading mode, four electrodes placed spatially around display	Position based on distance to electrodes or gesture based on nearest object to electrode
Transparent electric field sensor [LTI*14]	Transparent shunt mode array enabling near distance gesture interaction above mobile devices	Gesture Interaction	Shunt mode, four receiver electrodes	Positioning based on single proximity values and random decision forest learning

4.5.1. Indoor localization

One example system based on capacitive sensing is the previously presented TileTrack that uses a combination of transmit mode and center-of-gravity calculation between different floor tiles to calculate the position of multiple

persons [VMV09]. They are using special floor tiles comprised of chipboard and a steel layer. The steel layer acts as transmitting electrode that is coupled to persons moving above it. The initial system used wire and plate electrodes that were integrated into a wall and running to a height of 1.90m, thus ensuring receiving the field transmitted by tall persons. In order to properly localize a person the readings of multiple tiles relative to a single receiving electrode are combined. Similar to CapFloor (4.3.1) only passive materials are used below the floor level and the measurement systems can be hidden on the sides, however it requires large tile electrodes, thus it is mostly suited for hidden floor systems.

A second, already commercialized system is SensFloor that uses an integrated solution of capacitive sensors and wireless communication hidden below a floor covering that is able to detect the position of several users and other parameters such as falls, based on analyzing activity above single sensor areas or the movement trajectories over time [LS09]. The sensors work in loading mode and track activity in specific areas that are bordered by a sensing wire circle. Eight areas are attached to a single sensor, comprising a rectangular tile. The system is designed as a floor underlay, with power supplies being attached on the side and the single sensors communicating wireless to a base station. While this setup allows a high resolution, it is fairly expensive to build and potentially difficult to maintain.

4.5.2. Smart Appliances

Sato et al. have presented Touché, a swept-frequency capacitive sensor that allows distinguishing different types of touches on any suitable surface and medium [SPH12]. Some examples include recognizing different hand postures in liquids and touching different body parts to control mobile devices. Their system is based on analyzing a broader range of frequencies that have a different effect on the resulting capacitance. Using a classification method they are able to distinguish different categories of events. It supports an arbitrary number of electrode categories, yet only one single sensor attached to the measurement circuit. One additional feature is placing the electrode next to a conductive medium, such as water. Another example of this technology is touching different parts of a plant to control an interactive art installation [PSLS12], showing the versatility of this sensing technology.

HandSense by Wimmer and Boring uses four loading mode electrodes placed around a box mimicking a mobile phone to detect different grasping styles. The electrodes are made of tin sheet and sized 30mm x 15mm and placed on both small sides of the plastic box. Using different thresholds for "No proximity", "Near", "On hand", "Gripped" and "Held" they are able to classify six different types of holding the phone with varying accuracy. The basic idea is to use the grasping style as additional input device, e.g. by switching thumb-based interface items, when it is held in different hands.

4.5.3. Physiological sensing

Capacitive proximity sensors can be used to measure various physiological parameters that are related to movement of different body parts, including internal organs, most notably the heart. Cheng et al. have presented a system that allows measuring motions and shape changes of body parts using capacitive sensors embedded in garment [CAL10]. They combined four sensors based on loading mode that were exposed to electrodes made from conductive thread. They were attached to different body parts, including chest, wrist, upper legs and neck. According to position different forms of activities can be recognized, such as locomotion for the leg electrodes or swallowing for the neck electrodes. Additionally, the neck system could be used to detect the respiratory rate for some individuals.

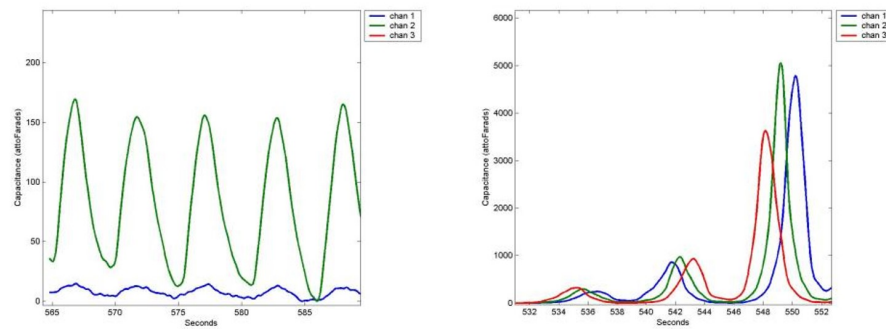


Figure 4.60.: *Left*: Spread spectrum capacitive sensor detecting respiration 30cm (green) and 90cm (blue) from chest. *Right*: Same sensor detecting person walking by at 100cm (left), 70cm (middle) and 35cm (right) [Mac04].

MacLachlan presented a system that detects the respiratory rate of a person lying on a bed from a distance of 30cm using a single electrode and a highly sensitive sensing method based on spread spectrum methods that are commonly used in wireless communication [Mac04]. Using a single electrode of 14mm x 56mm and the coding method he is able to detect very small capacitance changes. Figure 4.60 shows on the left the changes in capacitance caused by chest movement at a distance of 30cm, respectively 90cm from the moving chest. The response of three different sensors to a person passing at a distance of 100cm, 70cm and 35cm can be seen in Figure 4.60 on the right. The system requires coding hardware next to the electrode, thus making it more complicated to manufacture, as opposed to the previous systems.

4.5.4. Gestural interaction

The School of Fish by Smith, is a shunt mode system that he developed in the scope of his PhD thesis and the basis for most of the capacitive sensor research at MIT in the 1990s [Smi99]. Comprised of a collection of hexagonal copper electrodes distinguished into senders and receivers. Larger arrays are able to detect position and orientation of two hands moving at a distance above it. Though it was not used for the school of fish the system could be easily placed below non-conductive materials, e.g. to integrate it into a table. The hands are modeled using a collection of spheres, enabling a calculation of the orientation.

Thracker by Wimmer et al. uses four electrodes placed around a display to detect the movement of a hand in proximity [WKBS07]. The electrodes are angled away from the screen to improve the resolution at a distance. Two different modes were evaluated. One tracked the position of the hand in front of the screen, while the second one investigated pick and drop gestures.

A recent work by Le Goc et al. is based on the dedicated GestIC capacitive gesture tracking microchip [LTI*14]. Using transparent electrodes in shunt mode they are able to precisely track the 3D position of fingers moving in the electric field. The electrodes are based on an ITO layer on acrylic glass placed above a mobile phone. A second ITO layer is used to shield from interference generated by the capacitive touch screen of the phone. They use a random decision forest method to detect the position of a finger.

4.6. Discussion

In this chapter I have identified several distinct use cases for capacitive proximity sensors in smart environments and several associated challenges. To overcome those challenges new methods have been developed that can be grouped into five categories. These areas are sparsely distributed sensor system, model-based approaches, heterogeneous sensor systems, image-based processing and physiological sensing. The methods have been implemented and evaluated using a set of six prototypes that are associated to one or more application areas. Additionally, I briefly described three additional prototypes that I created in collaborative projects and collected capacitive proximity sensors systems from literature. Within the prototype descriptions, the rationale for designing the sensor layout and the different processing methods are detailed. The combined knowledge gathered in the design process, allows me to evaluate the applicability of capacitive proximity sensors in smart environments. The next chapter will describe this evaluation that includes a comparison to other sensor technologies and the specific benefits and limitations of capacitive proximity sensors.

5. Evaluating capacitive proximity sensors in smart environments

In the previous chapters I have introduced capacitive proximity sensors and their potential applications in the domain of smart environments. I have created a number of prototypes and analyzed existing systems from literature. With this as a basis, I am now able to perform a meta-analysis to identify common patterns and identify benefits and limitations of the technology when applied in smart environments. This chapter is distinguished into four different parts. First, I will compare capacitive proximity sensing to other sensing technologies that are commonly used in smart environments. Afterwards, I will discuss the major benefits of the technology before outlining several limitations. Based on the collected knowledge, I will conclude this chapter with a set of guidelines that may help other researchers when designing smart environment applications based on capacitive proximity sensors. This discussion is extended from a paper currently in press [BWKF15].

5.1. Comparison to other sensing technologies

In order to properly place capacitive proximity sensing in the domain of smart environments, it is necessary to include a comparison to other sensing technologies. I have given an overview of common technologies in Section 2.3. The results are briefly recapitulated, in order to keep this section self-contained.

5.1.1. Overview of sensing technologies in smart environments

Capacitive touch sensing, as opposed to capacitive proximity sensing, relies on an surface being touched instead of an object being in proximity. It is ubiquitous in touch screen applications. The technology is actually very similar to capacitive proximity sensing, using the same measurement principle and similar electrode layouts. The electrodes are typically below a protective layer of non-conductive layer. However, they are tuned to detect very close proximity only. This restriction greatly increases the potential resolution in close distances.

RGB cameras are a class of image sensors operating in the same frequency domain as the human eye. They are capable of easily distinguishing different colors. Nowadays, the most common technologies are CMOS and CCD sensors that use specific filters to distinguish different wavelengths. They vary strongly in resolution and achievable frames per second. The major disadvantage is that they rely on external light sources and are therefore not well suited in dark environments.

Infrared cameras operate in near light frequencies that are invisible to the human eye. This allows for application in dark environments, as it is possible to project infrared light into the scene without disturbing the user. This is primarily used for illumination, but may also use particular patterns, the reflections of which can be analyzed. It uses the same type of sensors as RGB camera, while having different filters, resulting in a single channel image.

Ultrasound sensing uses low frequency range mechanical waves just above the audible limit of human hearing. The waves propagate similar to sound signals and one can perform reflection measurements or time-of-flight

Table 5.1.: Qualitative comparison between capacitive proximity sensors and other sensor technologies

Name	Application Domains	Environmental Influences	Detection Range	Processing Complexity	Unobtrusiveness
Capacitive proximity sensing	indoor localization, smart appliances, physiological sensing, gestural interaction	electric fields, conductive objects	near distance (< 100cm)	Few high dynamic range data sources	invisible integration possible
Capacitive touch sensing	smart appliances, physiological sensing, gestural interaction	electric fields, conductive objects	touch	Few binary sensors	thin cover above electrodes
RGB cameras	indoor localization, smart appliances, physiological sensing, gestural interaction	occlusion, external lights	far distance (> 10m)	Complex image processing based on resolution	pinhole lenses
Infrared cameras	indoor localization, physiological sensing, gestural interaction	occlusion, external infrared light	medium distance (< 5m)	Complex image processing based on resolution	infrared source and camera
Ultrasound sensing	indoor localization, smart appliances, gestural interaction	acoustic occlusion, absorbing materials	medium distance (< 5m)	Few low dynamic range data sources	emitter and senders with exposed pinhole speaker, microphone
Microphone arrays	indoor localization, smart appliances, physiological sensing	environmental noise, absorbing materials	medium distance (< 5m)	Very high dynamic range data sources	exposed pinhole microphones
Radiofrequency sensing	indoor localization, smart appliances, gestural interaction	other RF devices	far distance (> 10m)	Few low dynamic range data sources	hidden emitters and senders possible

methods. The emitters and receivers can be easily modified to use different frequencies, thus enabling to use a number of sensors in a single setup or reduce the amount of required receivers. The systems detect most materials unless they have a strong dampening effect on mechanical waves in that frequency range.

Microphone arrays detect signals in the range of human hearing, and thus work with audible signals, such as human speech. They are typically based on piezoelectric sensors that transfer the vibration of a crystal to an electric current. The frequency ranges are associated to human hearing, thus covering at least the range between 10Hz and 20kHz, but often also exceeding this. Common sampling rates include 48kHz or 96kHz.

Radiofrequency (RF) sensing uses signals in a range between several hundred kHz up to 5GHz, typically used for wireless communication. Commonly the signal strength or time of flight is used to gather information about the environment. This technology is very popular due to the prevalence of WiFi equipment in most modern environments, allowing to create applications based on existing hardware that can be additionally used for communication between different nodes. Other popular radio-technologies include ZigBee and Bluetooth.

A short overview can be found in Table 5.1. Going back to the benchmarking model, I have included a comparison of suitable application domains, environmental influences, detection range, processing complexity and unobtrusiveness of the technology. This subset of features is a selection of the set presented in Chapter 3 that were aggregated in Table 3.8. It includes more details on the particular aspects of the different sensing technologies related to the given feature.

5.1.2. Classification of capacitive proximity sensors

Most technologies are capable of supporting multiple application domains. Some non-intuitive examples include WiSee that enables whole-body gestural interaction, using WiFi signals [PGGP13], or TapSense that uses a single microphone to enable gesture interaction on various surfaces [HSH11].

Capacitive sensors are disturbed by conductive objects and electric fields, whereas cameras struggle with occlusion and external light sources. A line of sight is required, even though some materials, such as glass have different opacity properties for the different wavelengths, in this case blocking infrared light. Some plastics however will block visible light, while infrared passes through. Acoustic sensors are prone to dampening materials and environmental noise interfering with the signal. Similar to cameras a multitude of sensors can be used to filter out environmental noise or locate the source of a specific signal. Radiofrequency signals usually propagate well through most materials. There is a dampening effect, particularly for high-frequency ranges and weak signals. This results in a reduced reception of signals when indoors. Additionally, conductive objects may create a shield that significantly reduces the measurable effect of outside external fields. In the common WiFi frequency bands of 2.4 GHz and 5 GHz, a multitude of devices are communicating, leading to a limitation of the bandwidth and potential interference. This has to be considered when designing applications for this sensor category.

Regarding the detection range, mostly applications in buildings are considered. Here it is not usually necessary to measure properties of objects that are very far away, e.g. more than 10m. In this criterion, deteriorating sensor readings at a certain distance also have to be considered. One example is radiofrequency sensing that has the best range, but considerable limitations when being applied over short distances. The fast signal propagation makes it difficult to measure the time-of-flight without specialized hardware. Visible light cameras have to consider the required distance in design of the optics. However, it is possible to use adaptive optics to enable a higher range in suitable distances. Infrared cameras have deteriorating quality, particularly when using static pattern projection, limiting their range to a fairly short interval. Some additions include using stereo infrared cameras, or adaptive projection patterns. Ultrasound also gets less precise when exceeding a certain distance, due to the wave properties and single measurements. However, the slower propagation makes it suitable for applications at close distances. Capacitive proximity sensors struggle to measure objects that are far away from the sensing

electrode, as their influence on the electric field might be below the achievable resolution. Specific layouts of electrodes allow very precise measurements at small distances.

It is not trivial to find a good measure for the processing complexity associated to different sensing technologies. An approach is to take the dynamic range of a sensor and the number of sensors typically required. Dynamic range is the difference between the smallest detectable value and the largest detectable value. Microphones have a high dynamic range measuring over a large frequency scale, whereas touch sensors only have two different states. Ultrasound usually measures simple amplitudes in time-of-flight systems and thus does not require complex processing. Image processing in both visible and infrared domains requires complex operations on large data arrays and is therefore difficult to integrate in simple embedded systems. This discrepancy has a distinct influence on the number of sensor systems required in a system. A single touch sensor is required on each position that is supposed to be interactive. Using a single camera can account for watching a larger area, identifying numerous activities, or detecting fine activity information at a short distance. Capacitive proximity sensors have a high dynamic range for single sensors. However, as the electrodes need a certain size to enable detection at a higher distance, the number of sensors in a single setup is limited. Thus, unless complex processing methods are used the complexity can be considered low.

Regarding unobtrusiveness, capacitive sensors and RF sensors can be applied completely invisible, below any non-conductive material. Cameras, microphones and ultrasound need a direct line-of-sight or mechanical connection to the measured property and are thus more difficult to hide. As previously mentioned, certain plastics are transparent for infrared light and glass can be designed to provide varying transparency levels for different directions. Microphones can pick up any environmental noise and can be hidden below a surface that transmits mechanical waves well. One example is the CapTap prototype (4.3.6) that uses a microphone to detect acoustic events on a surface. Of the presented technologies, capacitive proximity sensing is the least obtrusive. The layer between the detectable object and electrode can be very thick, as shown in the Smart Bed system (4.3.2) that has a thick mattress layer.

The actual performance of a specific sensor is difficult to assess from this limited selection of criteria and often is highly depending on the selected application. Having worked with a number of different prototypes and analyzing the advantages and disadvantages of the different technologies it is possible to build a set of benefits and limitations of capacitive proximity sensing that will be presented in the following two sections.

5.2. Benefits

Given the information collected in the previous section I can now discuss the specific benefits of capacitive proximity sensing. I am using three different groups for categorization, namely versatility, unobtrusiveness and processing complexity. Some examples within these groups are shown in Table 5.2. In the following section I will discuss those groups in detail.

5.2.1. Versatility

A main benefit of capacitive proximity sensors is the versatility in which they can be applied. With flexible choice of electrode materials, size and geometry it is possible to create highly individual applications. Some example electrodes include transparent metal oxide layers, woven conductive thread, copper wires, PCB boards, or simple aluminum foil. In the different prototypes I have used a large variety of materials, including aluminum foil in the MagicBox (4.3.5), long copper wire electrodes in the CapFloor (4.3.5), conductive thread in the Capacitive Chair (4.3.5), and transparent ITO in GestDisp (4.4). Other materials in the literature include plant tissue for

Table 5.2.: Overview of capacitive proximity sensing benefits

Name	Examples
Versatility	Flexible electrode design, scalability, different sensing methods
Unobtrusiveness	Invisible application, non-disturbing frequency range
Processing Complexity	Small number of sensors, variable dynamic range

Botanicus Interactus [PSLS12] and water in Touché [SPH12]. The size and shape of the electrodes is varying accordingly. The finger tracking array in the Active Armrest (4.3.4) consists of 2 cm x 3 cm copper PCB, while the Capacitive Chair (4.3.3) has some electrodes of 15 cm x 20 cm. The largest electrodes of the related works can be found in the TileTrack prototype using 60 cm x 60 cm tiles [VRV10]. The electrode design can be modified freely according to the specific application scenario. Given the large variety of different sensing methods most scenarios can be covered.

Additionally, the sensor systems are also highly scalable. By choosing appropriate voltages and frequencies it is possible to add a high number of sensors to a single object. Rekimoto’s SmartSkin uses shunt mode, with nine emitters and eight receivers leading to 72 sensing points [Rek02]. The CapTap prototype uses three OpenCapSense boards, interfacing 24 sensors, relying on time multiplexing to prevent interference between the different sensors (4.3.6). The CapFloor prototype as used in the EvAAL competition connected 26 sensors to a single PC system, controlled by two different communication interfaces (4.3.1). Further iterations developed by Tobias Grosse-Puppendahl support more than one hundred sensors in setups covering multiple rooms. Using smart measurement windows and different multiplexing methods, sensors can be placed close together and electrodes may act as both sender and receiver, e.g. the frequency multiplexing method used in the Honeyfish system [GPB12] (4.4). The different sensing methods presented - loading mode, shunt mode and transmit mode, enable a variety of different sensing patterns. The human body can be used as both sender and receiver and smart electrode layouts allow using low-powered processors.

In conclusion, it is possible to add capacitive sensing to most everyday objects to enable different forms of interaction, create natural interfaces and smart objects. The developed prototypes are using different electrode materials, flexible or solid electrodes, conductive thread, wires, shielded or non-shielded layouts.

5.2.2. Unobtrusiveness

Electric fields are not usually perceived by persons, unless they are of exceptional strength. Capacitive proximity sensors usually operate with low voltages (OpenCapSense operates at 3.3V or 5V) and with frequencies that are not considered biologically active (OpenCapSense uses a 100kHz sampling rate). Furthermore, they propagate through many materials that are typically present in our environment, including most plastics, wood or tile. This allows to invisibly apply capacitive proximity sensors without a strong effect on the measurement. Regarding the Smart Bed, (4.3.2) the signal propagates through a thick layer of mattress, consisting of polyurethane foam. The CapFloor system is designed to work below wood, PCV, tile, or other non-conductive floor materials (4.3.1). An application below several centimeters of covering is possible, if the electrodes are designed properly for large distance sensing. The detection range of the CapTap demonstrator is not disturbed by the laminated wood tabletop of 5cm thickness (4.3.6). Similar the Capacitive Chair has electrodes attached below a cover comprised

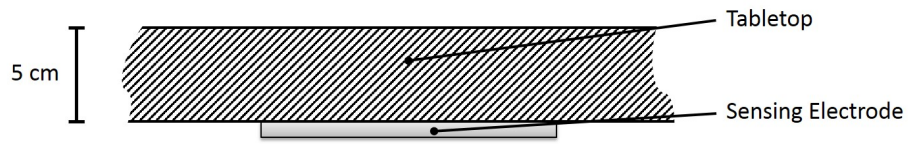


Figure 5.1.: Sketch of CapTap prototype showing thickness of tabletop

of wood, polyurethane foam and cloth, the materials of the seat area (4.3.3). Other materials that were applied above capacitive proximity sensors in the scope of the experiments were glass, polycarbonate, carpet, or access floors that are comprised of a wood, plastic and carpet combination. None of the materials we used showed unexpected behavior, related to absorption or attenuation of the sensor signal. If the sensors are attached close to conductive materials the signal can be disturbed if they are grounded or acting as shield towards the object to be sensed. Shielding in this context describes preventing the electric field created by the sensors to propagate to the object that is supposed to be detected, e.g. the famous Faraday cage. In some scenarios capacitive proximity sensors can also be used to detect properties of conductive objects, e.g. detecting the level of water in a glass or other non-conductive container.

It is possible to equip most conductive objects directly with capacitive proximity sensors, and hide them below non-conductive objects with minimal spatial requirements. The Active Armrest (4.3.4) prototypes are using sensor sets that are completely invisible from the outside. The CapTap was specifically designed to combine several sensing technologies that are invisible from the outside - hidden electrodes and a microphone (4.3.6). The thickness of the tabletop is shown in Figure 5.1. One disadvantage of this method is the disability to easily detect touch if there is no contact between object and electrode. Touching the electrode will typically cause a spike in sensor values that can be very easily discriminated. Therefore, in order to detect touches narrow thresholds and suitable sensing electrodes have to be used, as shown in SmartSkin [Rek02] or the touch area of the Active Armrest (4.3.4). Another potential solution is to use indirect effects of touching. The Smart Bed is using the pressure applied on the mattress to cause a geometric deformation of the electrode (4.3.2). This results in a clearly measurable spike in sensor values and can be modified for application below other flexible surfaces. The Smart Bed is additionally designed to communicate wireless to a PC, only using a regular power supply for operation.

5.2.3. Processing Complexity

An appropriate analogy to capacitive proximity sensors is a single photo diode. As opposed to a light intensity, capacitance is measured. While the information we can gain from such a measurement is limited, the processing required to analyze the signal is also low. Performing signal analysis on an array of 24 capacitive sensors, as in the CapTap prototype (4.3.6) is comparable to processing the image of a 6x4 pixel camera. Therefore, it is easy to create highly integrated systems with low-power devices for performing any subsequent data analysis. Other prototypes use even less electrodes, most notably the Magic Box that uses only six sensors and computationally inexpensive object localization algorithms (4.3.5).

While it is in many cases beneficial to use complex data processing algorithms for object detection, it is often possible to achieve a similar result with less complex methods. In many applications it is even viable to opt for a

quantized capacitance measurement. In the case of a touch sensor a single binary measure is sufficient. However, it is also possible to select various different levels. The Active Armrest processes the two back electrodes in a quantized fashion, just using a single threshold for each (4.3.4). The sensor dynamic can be reduced to any suitable range, thus reducing the processing complexity later in the algorithms. The CapFloor system uses a larger number of sensors and does not require a high dynamic range for calculations (4.3.1) and thus uses shortened 4 Bit values.

With the exception of the Capacitive Chair (4.3.3) and the CapTap (4.3.6) that make extensive use of frequency domain operations, our prototypes are using simple data processing methods that can be easily applied on embedded systems. One of the presented examples is the weighted average algorithm for object detection, as shown for the Magic Box (4.3.5). Regarding model-based data processing, even very simple cylindrical models, such as the one used for the Smart Bed (4.3.2), are capable to reliably predict numerous postures that are relevant in real world applications. In general, the low requirements for data pre-processing, allows dedicating more resources to high-level data processing algorithms if the specific application is resource constrained. Thus, it is easily possible to fuse the data of multiple heterogeneous sensors in the system or apply more complex classification methods, such as the SVM classifier used for the Capacitive Chair (4.3.3).

Another option to reduce the processing load is reducing the sample rate of the sensors. If this is done in hardware, there is an additional benefit to this method. The sensitivity of the sensors can be increased, as loading and unloading times can be decreased, making it easier to detect changes in the resulting capacitance. However, this method is not feasible for all electrode sizes, as particularly small electrodes will reach their capacitance limit early. Another option to increase the sensitivity is increasing the voltage, which influences the propagation range of the electric field. In this case, it is more difficult to guarantee biological inactivity and a low power consumption.

The OpenCapSense toolkit that is the base for most of the prototypes has a fairly powerful micro controller that is able to implement all of the processing steps. This enables highly integrated, low-power capacitive proximity sensing prototypes that can be used in smart environment applications. It is also viable to use low-powered PC systems, such as the Raspberry Pi, to interface a large number of capacitive proximity sensors in research applications.

5.3. Limitations

Despite the potential benefits that have been outlined in the previous section, capacitive proximity sensors have a number of limitations that may hinder several applications. Similar to the benefits I am putting these into three different groups that are outlined in Table 5.3. Environmental influences have been briefly mentioned throughout this work and have to be considered in any setup. The physical range limits the object detection to a low distance from the sensing electrode. The objection detection similarly suffers from the physical limitations of electric field sensing. These three groups will be detailed in the following paragraphs.

5.3.1. Environmental Influence

One of the main limitations of capacitive proximity sensors is their sensitivity towards environmental influences. Any factor that modifies an electric field will also affect the measurement of a capacitive sensor. The current environmental parameters, like temperature and humidity are having a considerable effect on the medium in which the electric field propagates. However, those changes are usually over a longer period of time and can be compensated using a factor for drift, as described in the previous sections about noise reduction. While the frequency

Table 5.3.: Overview of capacitive proximity sensing limitations

Name	Examples
Environmental influence	Static electric fields, dynamic electric fields, temperature, humidity, conductive objects
Physical range	Small differences in capacitance, reduction due to influences, physical limitations
Object detection	Small number of data points, a priori knowledge

range in which the sensors operate is usually not in an interval that disturbs other electronic systems, there are some electromagnetic sources in the environment. The most common are power supplies (frequency range of 50-60Hz) terrestrial radio (frequency range of 150kHz to 100MHz), mobile phone communication (870-970MHz and 1700-1900MHz), as well as WiFi (2,4GHz and 5GHz). While in theory there is some potential interference with the terrestrial radio spectrum, the effect is fairly small and can be covered by choosing intermediate frequency intervals. Thus it is feasible to use capacitive sensing even in environments, where non-disturbance is a main requirement.

The main source of disturbance occurring in our evaluations, was the influence of very close electromagnetic sources and disturbing signals within the power grid. One example for a disturbing source was a plasma TV installed in our Living Lab. When turned on the CapFloor prototype (4.3.1) suffered from severe noise, preventing any successful tracking of persons. The culprit are high frequency internal power supplies that drive the plasma cells. Newer TVs are using different internal electronics that significantly reduce the emitted electric field. Another disturbing factor can be faulty power supplies that are attached to the grid and produce frequency noise. As this noise typically operates in a limited frequency range, it is possible to use frequency domain filtering that attenuates these bands. Thus, it should be attempted to apply capacitive proximity sensing in an environment that adheres to all relevant standards regarding electromagnetic compliance. Additionally, the systems should be designed in a way that reduces potential external influences, e.g. by using internal power supplies that are stabilized and applying shielding where necessary.

An additional issue might arise when placing sensors close to each other. The created electric fields may disturb the measurement if some electrodes are charged and create fields to adjacent electrodes while those are discharged for measurement. This is particularly challenging, when a larger number of sensors is involved. Here, this is the case for the CapFloor (4.3.1) and CapTap (4.3.6) prototypes. In the case of CapTap specific charge-discharge cycles for the electrodes are used that ensure that adjacent sensors are not interfering with each other. Similarly the CapFloor attempts to control the sensors in a way that prevents interference, taking into account the geometric layout and the position of the electrodes within the area. Both methods are a form of multiplexing that is based on time (TDMA). Other options include a variety of different multiplexing methods, such as the previously mentioned frequency method of Honeyfish (4.4). Code division multiplex is a last class of multiplexing methods that relies on the modification of the signal. This was employed by the presented capacitive sensor of Rob MacLachlan [Mac04]. It can result in very precise measurements, but requires specialized hardware. Thus, it was not used in any of the developed prototypes.

Another major challenge are conductive objects that are permanently placed in the immediate sensing environment. It is difficult to distinguish the object we want to detect from a disturbing object, if their influence on the electric field is similar. Long term data analysis may help in performing a successful detection. The CapTap

prototype, as a regular living room table, should not fail when disturbing objects are present, such as glasses full of water (4.3.6). This can be achieved by performing a feature analysis that detects the typical response of the disturbing object and performs a recalibration that takes the object into account.

The CapFloor prototype is affected by environmental influences the most, given the small size of the electrodes relative to the interaction area and the changing environment on top of the floor (4.3.1). Strong noise reduction algorithms and drift compensation are used to create a more stable result that however causes a reduction of the detection range.

5.3.2. Physical Range

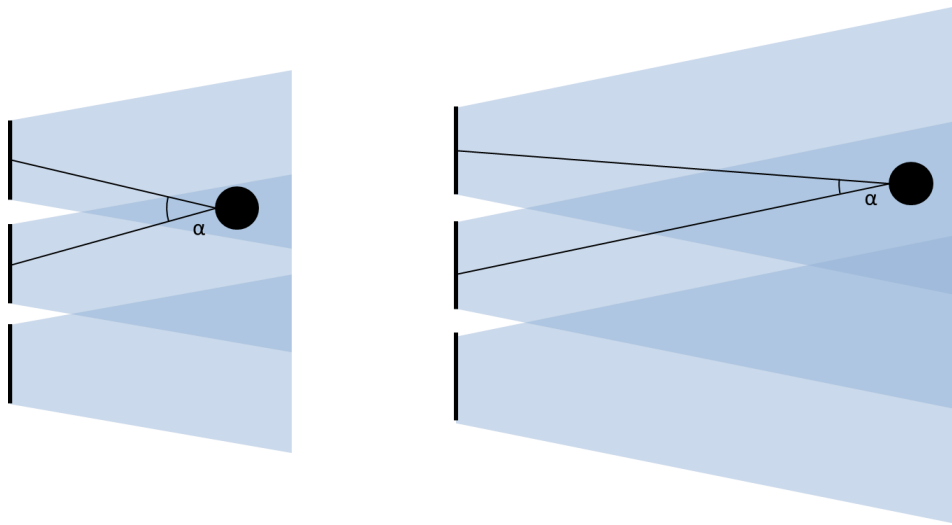


Figure 5.2.: Reduced angular resolution on smaller, distant objects

The physical range of the generated electric field is one of the main limiting factors of capacitive proximity sensing. In order to detect objects that are further away, the electric field strength has to be increased sufficiently. This is easier for larger electrodes, as its potential capacitance is higher. However, this also leads to distant objects having an ever smaller influence on the overall capacitance. In this case we need more precise measurement circuits and longer measurement times to improve the signal-to-noise ratio. Additionally, looking at smaller objects the angular resolution will decrease as shown in Figure 5.2. This makes it more difficult to get a precise localization as the immanent noise leads to an angular error. While this can be compensated using more sensors, the far distance would require us to use large electrodes that have to be placed further apart resulting in a huge area that would have to be equipped with sensor electrodes. The Capacitive Chair prototype uses large electrodes that allow us to detect the presence of a person at distances of around 50-60cm (4.3.3). It is designed to detect whole body parts and thus does not need any fine resolution. The Active Armrest on the other hand has a finger interaction array comprised of small electrodes that have a detection limit of approximately 15cm (4.3.4). Yet, it is able to detect fine movements and precise locations of the fingers within this range.

In general, the achievable resolution is not comparable to vision based system and has to be taken into consideration when designing the specific application. A balance between electrode size, physical range and achievable resolution has to be found. The MagicBox size does not allow an integration of very large electrodes (4.3.5).

Instead, we are optimizing the available space in order to achieve a detection that lets us detect hands in a distance of approximately 30cm and allows reliable tracking in distances between 15cm and 20cm. The system with the lowest detection distance is the CapFloor prototype (4.3.1). The long wire electrodes are not particularly well suited for achieving a good resolution and are prone to interference. However, they are suitable for this specific purpose and objects at a proximity of 10cm or less can still be detected.

5.3.3. Object Detection

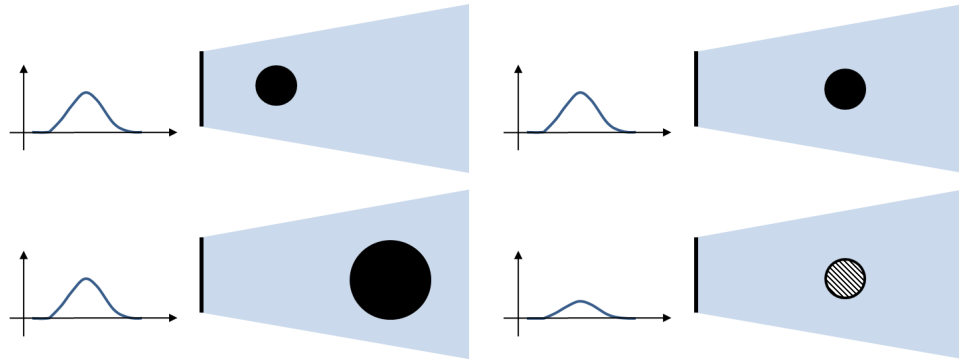


Figure 5.3.: Same response to differently sized objects (left), different response to varying materials (right)

Object detection using capacitive sensors, can be partially compared to object detection using camera systems, with a single sensor being equivalent to a single photo sensor. Smith et al. propose that loading mode measurements resemble light captured by a camera without a lens, as only one part of the electric system is constrained [SWD*98]. The light intensity measure is comparable to field intensity and likewise it is not possible to distinguish if the measurement is caused by a weak source in close proximity or a strong source at a further distance. As a practical example the capacitive sensor is not able to decide if one hand is close to the sensor or two hands are a bit further away. This effect makes it challenging to provide object detection and usually information from various sensors has to be combined to get a good idea about object shape and size. The CapTap prototype is able to detect two hands by combining the information of 24 sensors and additionally uses an analysis of the arm posture in the detection area (4.3.1). Due to the presented challenges in physical range and electrode size, capacitive proximity sensing systems do not have the same level of scalability as cameras, where millions of photo sensors can be placed in very small areas. Touch screens show that capacitive sensing technology has a very high resolution in near ranges. However, this can't be translated to similar resolution in further distances.

Additionally, the effect of an object on the electric field is not always closely correlated to the object dimensions, but instead based on conductivity, material and other factors. The same response to different objects may be measured at different distances, if they are differently sized or are made of different materials, as shown in Figure 5.3. In the presented application scenarios exclusively parts of the human body are tracked. Mostly, there is also an assumption that no other objects should be used. The CapTap will have to identify potential foreign objects from the hands of the user (4.3.6). In this case, the specific capacitance profile of objects at close proximity can be used, similar to the capacitance tags presented by SmartSkin [Rek02]. The Active Armrest has gestures for one and two fingers that are distinguished using a simple threshold (4.3.4). If another object is entering the field, or the person is having a strongly different finger size, the system will fail to properly differentiate gestures. Accordingly, some other compensation methods should be used.

5.4. Guidelines

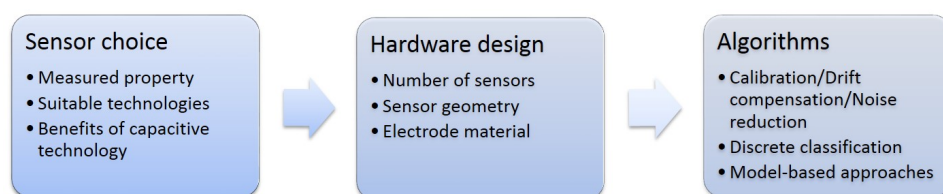


Figure 5.4.: Basic system design process for capacitive proximity sensing systems and associated aspects in the single steps

After discussing the limitations and benefits of capacitive proximity sensors, I am able to use the collected experiences and knowledge to give some general guidelines that can help parties interested in using this sensing technology. These guidelines will support the design process from sensor choice, to choice of algorithms. The process is shown in Figure 5.4. This section is structured following the three design steps sensor choice, hardware design, and algorithms. In each subsection a set of questions are posed that are associated to the specific challenges. These questions will be answered with references to the different prototypes created in the scope of this work, and how it influenced the design process.

5.4.1. Sensor choice

The first step of this process is the choice of sensor technology for the given application. There is a variety of different options, as presented in Section 5.1. All have their different strengths and weaknesses. In order to decide if capacitive proximity sensing is suitable, three different questions should be posed.

What do I need to measure in my application scenario?

Capacitive proximity sensors can measure the presence and properties of conductive, grounded objects. A prominent object of this category is the human body, having distinct electrical properties. This includes the various application scenarios shown in the previous sections. However, if the application requires measuring properties of unsupported objects that are non-conductive, a different technology should be chosen. In some specific scenarios non-grounded conductive objects can be measured if they are coupled to the electrode. One example is measuring the fluid level of a non-conductive glass container placed on an electrode.

What sensing technologies are supporting the required measurements?

It may be the case that multiple technologies support the measurements required in this specific applications. Cameras often can provide similar recognition as capacitive sensors, e.g. in indoor localization applications. In this step all potential sensing technologies should be collected. Additionally, the requirements of the application and constraints towards the environment have to be taken into account. If there are requirements on the visibility of sensing devices, or potential interference due to the unique limitations of a sensor technology, they can be discarded in this step.

Are capacitive proximity sensors beneficial for my scenario?

An evaluation of the different candidates is the final step and should lead to a decision about the most suitable sensing technologies. If the distance is too high for capacitive proximity sensors or enough processing power is available and lighting conditions are static, cameras might be more suitable. This should be driven by the different benefits and limitations of the technologies. The benchmarking model presented in Chapter 3 can support this decision process. The designer will have to find the respective feature weights for his applications and calculate the scores for the different sensing systems.

5.4.2. Hardware design

If there has been a decision in favor of capacitive proximity sensors the next step is the design of the hardware. Similar to technology selection we can use a few basic questions to get an idea of what layout to use.

How many sensors are required to get the measurement?

The number of sensors required is depending on the area to be covered, as well as the specific object parameters and the resolution that should be achieved. There is some limitation to the maximum size of the electrodes, as a single sensor can only charge a maximum capacity. However, this theoretical limit is never tested in any of the prototypes presented in this work. If a large area has to be covered more electrodes and sensors are typically used, as the technology is cheap and large electrodes will limit the available resolution. If we just want to measure the presence of a hand a single electrode may suffice. If orientation and position are interesting we need to combine measurements from various sensors.

We used six electrodes for the MagicBox (4.3.5) as it is only tracking a single hand on a limited surface. Most prototypes (4.3.2, 4.3.3, 4.3.4) use eight sensors as this turned out to be a good compromise providing sufficient data that can be collected by a single sensor kit. CapTap (4.3.6) uses 24 sensors as we wanted a high resolution over a larger area and CapFloor 4.3.1 is scaled to room size and may use a high number of sensors.

What should be the size and geometry of the electrodes?

This is closely related to the previous question. If the application is not restricting the available space, the electrode should be approximately of the same size as the object that is to be detected. This generates the highest difference in capacitance when the distance is changing. The Active Armrest (4.3.4) uses six very small electrodes for finger tracking and two larger ones for detecting the arm presence.

CapFloor (4.3.1) uses wire electrodes that are spread throughout the room to maximize coverage. The hand tracking devices (4.3.4, 4.3.5) are equipped with hand-sized electrodes and the body sensing devices (4.3.2, 4.3.3) use as much of the available space as possible.

What is the best electrode material to use?

Copper is always a good first choice as electrode material. If elasticity is necessary a thin sheet of copper, tin or aluminum can be used. For rigid structure solid PCBs are a good choice, as they also include a copper layer that may be used for shielding. In applications that require transparency, one of the presented materials will have to be used, such as ITO or PEDOT:PSS. Some touch screens also opt for very thin wires that are not visible in typical use with sufficient backlighting. For some applications it is beneficial to integrate electrodes into cloth. In this case there is a variety of conductive threads that may be used. In recent years, printing electronics on paper as become available for home and research applications [KHC*13]. Any conductive material will act as an electrode, thus the application and budget should be the primary driver of this decision.

Active Armrest and CapTap (4.3.4, 4.3.6) are not con-strained and use solid copper electrodes. CapFloor (4.3.1) uses copper wires. MagicBox (4.3.5) integrates the sensors into a thin casing, thus we used aluminum foil. As we wanted the electrodes of the Smart Bed (4.3.2) to deform along with the slatted frame we used copper

foil. The Capacitive Chair (4.3.3) uses a number of solid copper electrodes but also includes a single conductive thread, as it could be integrated into the mesh structure of the back rest.

Does my application require any shielding?

Shielding allows detecting only objects approaching from a certain direction. If the application requires this additional hardware, because it is anticipated that other objects might disturb the measurement, shielding should be used. The only prototype that uses shielding is the CapTap (4.3.6), as various electronic devices are integrated into the table and we wanted to minimize disturbance. Shielding however, requires hardware support and comes at additional cost and complexity.

5.4.3. Algorithms

Finally, if the hardware is designed as desired, the different variations of data processing algorithms have to be chosen and configured according to the application.

What pre-processing algorithms should be used?

Using baseline calibration is beneficial in the vast majority of applications. Having a distinct starting point simplifies all further steps of high-level data processing, such as normalization and setting different thresholds. This step may only be omitted in very stable environments and if the system has sufficient prior information to operate on raw data. Drift compensation should be handled in a similar fashion. The common methods are not computationally expensive and having a stable baseline over time allows the same algorithms to be applied in a more robust fashion. The method and configuration of noise reduction are strongly depending on the specific case. Some form of noise reduction might be required in most applications. Yet, according to the type of noise different methods can be used. If outliers are an issue a median filter is appropriate; if a smoother signal is desired an average filter can be used.

In most of the presented prototypes an adaptive baseline calibration is used, together with dynamic normalization, capturing on-the-fly maximum and minimum values. Noise reduction relies mostly on an adaptive average filter that uses a varying sample count based on the extent of changing sensor values, which allows the filtered signal to react more quickly to events.

What are suitable methods for high-level data processing?

Regarding high-level data processing there are numerous variations of methods presented in this work. Data-driven machine learning algorithms are a good method, if there is a small set of potential outcomes, e.g. the different postures that could be recognized on a chair or couch. We partially use these methods in the Smart Bed (4.3.2), on the posture classification of the Capacitive Chair (4.3.3) and for the gesture recognition of the Active Armrest (4.3.4). Of the related prototypes most notably Touché uses these classification methods [SPH12].

If the application has many different potential outcomes, e.g. the thousands of potential locations in a hand tracking system, it is typically beneficial to use a model-driven approach, used in all other prototypes (4.3.1, 4.3.4, 4.3.5, 4.3.6). However, these models may be supported by data-driven algorithms, such as particle filters. One example is the Swiss-Cheese object tracker by Grosse-Puppendahl et al. [GPBKK13]. The data processing examples, shown in Section 4.2, give an idea of the decision rationale in various application domains.

5.5. Discussion

In this chapter I have evaluated capacitive proximity sensors in the domain of smart environments. As a first step of this evaluation, I provided a comparison to other sensor technologies. These were chosen based on their frequent use in smart environment applications.

This build the base for discussing the specific benefits of capacitive proximity sensors. They have a high versatility, based on flexible electrode designs and a number of different sensing modes. They can be applied unobtrusively, as the electric field propagates through any non-conductive object. They have a low processing complexity, as the number of sensors is small compared to other technologies. However, there are several limitations of capacitive proximity sensors. They are prone to environmental influence and drift over time. The physical range is limited to a few meters in best case scenarios. They struggle in detecting certain categories of objects, as the number of measurements is limited and different materials have a distinct response to electric fields.

Based on this classification I provided a set of guidelines that support parties developing smart environment applications throughout the development process, from choosing a sensor technology, over designing the hardware, to selecting suitable algorithms for processing the data.

6. Conclusions and Future Work

With this work I set out to evaluate the applicability of capacitive proximity sensors in the domain of smart environments. Technological advances in the last decades have enabled research and industry to create numerous devices and applications that aim at making our environments more aware and provide a variety of services. When designing such applications the developers choose from a selection of sensor devices that provide environmental measurements or activity data. Capacitive proximity sensors have some unique properties that are beneficial in scenarios that require unobtrusiveness. To support the sensor selection I provide a tool that allows to benchmark different sensor technologies against one another given a set of features and potential applications. I establish a set of application domains that are particularly well suited for capacitive proximity sensors. Each of those domains has a specific set of challenges regarding the required sensor technology, layout of the systems, and processing. These can be tackled using a variety of new or improved methods. In order to establish a classification of capacitive proximity sensors and its competing technologies, it is necessary to provide a set of implementation guidelines. This should be based on knowledge gathered from designing and evaluating a comprehensive number of prototypes.

I have developed a benchmarking model for sensor technologies in smart environments that is adapted from previously presented benchmarks for pervasive systems. This model is modified to include specific metrics for smart environments. The benchmarking score is calculated, by taking user-specified ratings as a weighting factor for the sensor characteristics. The method was evaluated by measuring the popularity of four different sensor technologies, used in three different application domains. For this evaluation I queried two scientific databases. Additionally, I provide two extensions. The first compensates the preference of some users to prefer average ratings. The second normalizes the final score if the user ranking result in different overall ratings. Thus, I was able to identify application domains for capacitive proximity sensors by choosing the four most suitable out of a set of seven.

Indoor localization, smart appliances, physiological sensing and gesture interaction are use cases that have been realized using capacitive proximity sensors in the past. However, the data processing still presents a number of challenges that can be overcome by improved methods. Next, I will list my specific contributions and the challenges they address. I used gesture by example to improve gestural interaction if the number of sensors in an interaction device is constrained. I developed an electrode grid layout that detects location and falls in rooms, with a limited number of sensors attached on the sides. Using abstraction I create analytical models of the capacitance measurements in different scenarios. A single body model is sufficient to precisely determine occupation and posture on a bed, while multi-body models on a chair allow a fine-grained calculation of the posture. Heterogeneous sensor systems can augment the range of applications or alleviate limitations of a single sensor type. I present a method to detect the intention to interact based on limb postures. Thus I can integrate capacitive interaction devices in areas where unintentional gestures can occur. A combined capacitive, acoustic system that detects different types of touch, increases the versatility of invisible capacitive interaction devices by providing additional input channels and a more reliable detection of selection events. Image-based processing of capacitive proximity sensor data primarily has been used for shunt mode systems. I present a method for loading mode systems, which can be applied more flexibly. It tracks the position and orientation of two arms in proximity of a table, using a selection of typical image processing methods. Finally, there are two methods for physiological sensing that detect respiratory rate based on chest movement, and sleep-phases based on whole-body movement.

Operations in frequency space are particularly well suited, because they capture periodic signals generated by muscle movements of the body.

In order to evaluate those methods and gather sufficient information to classify capacitive proximity sensors, six different prototypes are presented in detail. Each prototype is associated with one or more of the specified application domains. I outline the development of the prototypes, including the rationale for the selected electrode layouts and the processing methods chosen. I describe how specific design challenges are solved in order to build a knowledge base for specifying and detailing the capabilities of the technology. For all prototypes some form of functional or qualitative evaluation has been performed, with setup and results described in the respective sections. All prototypes performed well in their specific application scenarios. However, a limitation of these evaluations is that it is difficult to directly compare results to other sensing technologies or similar prototypes, as the data sets and sample users vary across different setups. Three additional prototypes based on capacitive proximity sensors that have been created in other collaborative projects are briefly introduced.

The knowledge gathered from designing the prototypes and evaluating their functionality allows me to build a base for classifying the technology and comparing it to other sensor technologies. Capacitive proximity sensors are uniquely affected by conductive objects and disturbing electric fields in the vicinity. However, they do not require a line of sight and optical and acoustical properties of the environment do not matter. The detection range is lower than most other sensing technologies, particularly radio frequency systems. However, the resolution compares favorable in low distances. The processing complexity is considered low, due to the low number of sensor values that occur in typical applications, despite the high dynamic range a single sensor can achieve. However, in some instances more complex processing methods are beneficial. An example is the SVM classification methods used in the multi-body models. Capacitive proximity sensors can be applied completely invisibly below non-conductive materials. This is also true for radio frequency sensing, but not true without limitations for the other sensor systems. In the prototypes a variety of materials including plastics, cloth, leather, and wood, are applied in a thickness of up to 5 cm.

A major limitation of capacitive proximity sensors is the proneness to certain types of environmental influences. Induced external and internal electric fields and changing parameters within the environment can cause a disturbance of the sensor signal. One example is periodic waves on the power supply lines that were resulting in unusable sensor readings. This problem can usually be mitigated by proper system design. However, in settings that have an expected permanent influence of electric fields some other sensing technology might be better suited. The physical range of the sensors is fairly low, particularly compared to cameras. As capacitance is an aggregate value of a local electric field, the information we can gather from a single measurement is limited. This primarily applies to distinguishing objects at a distance. However, constrained low-distance systems in industrial applications can be used for material detection, relying on the distinct dielectric properties.

Based on the collected knowledge in identifying suitable application domains, creating novel methods for data processing and building a variety of different prototypes, I created a set of guidelines that supports the design process. They support the selection of suitable sensors, choosing a viable hardware layout and reasonable processing methods. These guidelines are intended to support other researchers in developing systems based on capacitive proximity sensors that may be new to this technology.

I conclude this section by discussing some potential limitations of the work presented. The developed benchmarking method is based on a set of user-supplied ratings and therefore is subjective. Given the plethora of potential sensor variations, algorithms and applications it is difficult or even impossible to find a set of features and ratings that are universally valid. This benchmarking model is not intended to be used in a very generic fashion, but instead has to be applied to a specific set of applications. In some instances the frequency of keywords in the technology popularity evaluation is low, making it difficult to estimate the significance of the technology application combination. So far, the method has only been used on smart environment systems. Therefore, it is

not obvious how straightforward a transfer to other domains would be. Even in the domain of smart environments only a limited subset of application domains has been investigated. In order to identify novel use cases the method would have to be evaluated on a larger scale.

6.1. Future Work

Capacitive proximity sensing has a promising future in smart environments. Further benchmarks performed using the presented model can identify novel use cases. Unobtrusively detecting the human body to recognize activities or create interactive surfaces might alter the way in which we interact with our environment. Household appliances and furniture that reacts to presence, mood and activities can support energy saving services based on localization, or sleep and wake up support based on real-time detection of the sleep quality. Capacitive proximity sensors have a broad range of smart home applications and can be applied flexibly, by both designers and engineers. Increasingly, physiological measurements are used to detect activities and adapt systems. Today, this sensing relies primarily on wearable devices, such as smart watches. Remote measurement of the breathing rate has been implemented in the past. With newer hardware and more sophisticated processing, it could be possible to also measure heart rate, or even create some kind of ECG without physical connection to the person. This more detailed contextual knowledge about the persons in the environment can be used for a variety of applications, for example in the domain of personal health.

The scope of smart environments has already been expanded and includes vast public spaces, but also more constrained spaces. Millions of people are spending hours each day in their vehicles, on their way to work, or visiting friends and family. I have presented two systems that enable novel forms of interaction for cars. GestDisp detects touch gestures in front of a screen and the Active Armrest registers arm and hand position to detect the intention of a user to interact on an armrest. Capacitive proximity sensors can be used to create a variety of interactive surfaces in a car. It is feasible to reduce the number of buttons currently installed on many steering wheels by replacing them with interactive surfaces that are invisibly integrated. Sensing the physiological properties of the driver can reduce the number of accidents caused by unaware drivers. Sensors in the seat can register the onset of drowsiness or increased stress levels and could suggest a pause for the driver. Likewise, posture and occupancy sensing, as used in the Capacitive Chair, can verify that a suitable seating position is set, or adapt climate controls and multimedia systems according to number of occupants and their preference.

Benefits of capacitive proximity sensors include the flexibility of electrode layouts and designs, as well as their applicability on large surfaces. The combination of these properties could enable the creation of interactive surfaces that are irregularly shaped. This kind of interaction system is particularly interesting in collaborative work environments. The mixed form of free-air and touch interaction supported by systems such as CapTap can be combined with virtual or augmented reality to create immersive systems for multiple users. It is feasible to create curved surfaces that are designed to more naturally follow the movements of the human arm.

Finally, the benchmarking model can and should be expanded or adapted to include other domains outside of smart environments. This may require the omission or addition of features. These areas should support and use different types of sensor technologies. One candidate area might be robotics.

Acknowledgments

While many consider writing a PhD thesis to be a mostly personal endeavor, there are always various sources of discourse, collaboration, support and inspiration.

I would like to express my gratitude towards Professor Dieter W. Fellner for the guidance and support I have received over the last years. Likewise, I want to thank Professor Mühlhäuser for agreeing to evaluate this work and providing valuable input. Dr. Arjan Kuijper and Dr. Reiner Wichert have accompanied my work over the last years and have been a steady source of constructive criticism, review and new ideas.

I would like to thank my colleagues and frequent co-authors of the competence center for Interactive Multimedia Appliances at the Fraunhofer Institute for Computer Graphics Research IGD for discussing and sharing ideas that shaped this work. A special thanks goes to Pascal Hamisu, who introduced me to the magical world of capacitive sensors.

Without the excellent work performed by different Bachelor and Master students I had the opportunity to supervise over the last years this work would not have been possible. They are in chronological order Arthur Fischer, Tobias Große-Puppenthal, Martin Majewski, Yannick Berghöfer, Stephan Neumann, Stefan Krepp, Sebastian Frank and Sebastian Zander-Walz.

Likewise, without the excellent help from numerous student workers this work would not have been possible. So my thanks additionally goes to Henning Heggen, Sönke Schmidt, Mark Prediger, Steeven Zeiß, Maxim Djakow, and Pia Weiß.

The patience, encouragement and support of my parents Margit and Werner, as well as the rest of my family has been immense in the last five years. I am eternally grateful.

Kathleen, you kept me grounded, cheered me up when necessary and removed some of the worst offenses to the beautiful English language. I will be forever grateful for your love.

A. Evaluation results

This appendix will give an overview of the raw results gathered by the evaluations performed on the different prototypes.

A.1. Capacitive chair posture recognition

This section provides raw data of the first posture recognition test performed using the capacitive chair.

A.1.1. Evaluation setup

Short study with 10 participants. Three poses and a non-pose:

- Sitting upright
- Sitting hunched
- Slouching on chair
- Close to chair - disturber

The persons were given a short introduction, the different postures were displayed, and finally the persons were asked to perform the postures in order. When testing “close-to-chair” the subjects were asked to rattle at the chair, stand close, move it around and thus disturb the potential sensor readings. Each class was tested for 10 seconds, collecting 200 samples.

A.1.2. Raw results

Table A.1.: Percentage of correctly classified postures using manually set classifier											
	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	Avg
Upright	100	100	100	100	100	100	100	100	100	100	100
Hunched	100	100	100	100	86	100	100	100	100	100	98,6
Slouch	100	100	100	100	100	100	100	100	55	100	95,5
Disturber	100	100	100	100	100	100	100	100	100	100	100

A.1.3. Postures

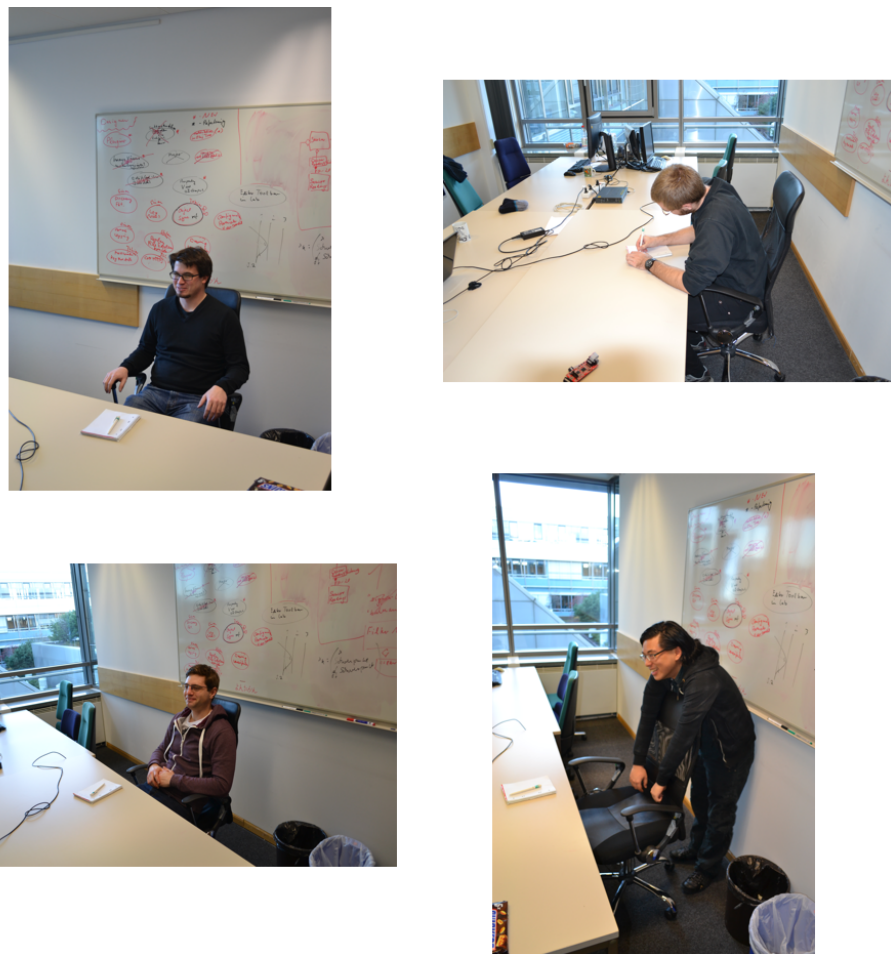


Figure A.1.: *Top left:* Upright posture. *Top right:* Hunched posture. *Bottom left:* Slouched posture. *Bottom right:* Disturber posture

A.2. Capacitive Chair - working situation recognition

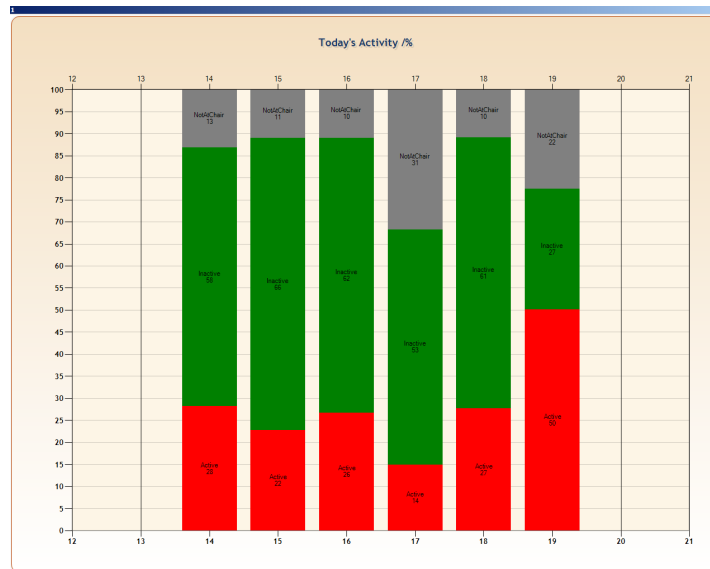


Figure A.2.: Work activity day 1

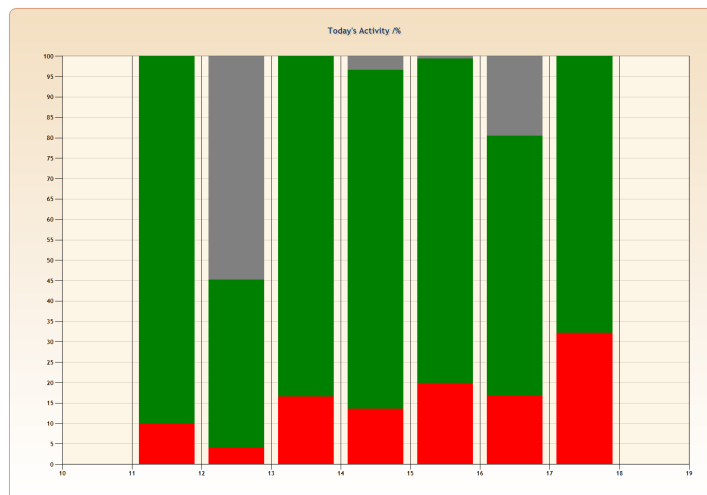


Figure A.3.: Work activity day 2

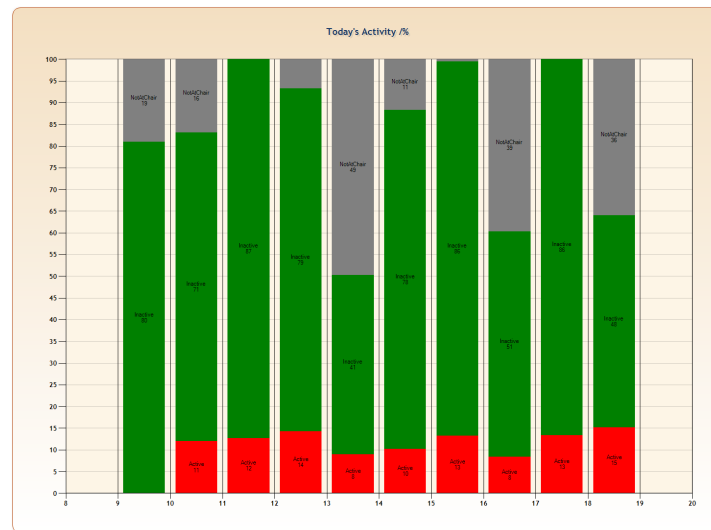


Figure A.4.: Work activity day 3

A.3. CapTap evaluation

A.3.1. Questionnaire

- G1 Experience with touch screen systems (none = 1, daily usage = 10)
- G2 Experience with gesture recognition systems (none = 1, daily usage = 10)
- Q1 Do you agree that the required tasks were easy and precise to perform? (Not agree = 1, Strong agree = 10)
- Q2 Was the CapTap to be intuitive in its usage? (Not agree = 1, Strong agree = 10)
- Q3 I could control the different layers in the painter application? (Not agree = 1, Strong agree = 10)
- Q4 I could control the different interactions in the test run? (Not agree = 1, Strong agree = 10)
- Q5 Do you prefer finger swipes or hand swipes? (Finger swipe = 1, Hand swipe = 10)
- Q6 Do you prefer finger taps or knuckle knocks? (Finger taps = 1, Knuckle knocks = 10)
- Q7 The CapTap should support more than three interaction layers? (Not agree = 1, Strong agree = 10)
- Q8 Do you agree that CapTap is an interesting form of interaction device? (Not agree = 1, Strong agree = 10)
- Q9 Would you consider using an input device like CapTap for a longer period of time? (Not at all = 1, I would like to use it = 10)
- Q10 What did you particularly like about the CapTap?
- Q11 What did you particularly dislike about the CapTap?

A.3.2. Raw results

The table denotes the results of the different touch points as explained in the descriptive section. The T's refer to the times of the three different interaction speed runs.

Table A.2.: CapTap evaluation raw results

Subject	1	1D	2	2D	3	3D	4	4D	5	5D	6	6D	7	8	9	10	11	12	13	14	15	T1	T2	T3
S1	3	3	3	3	3	3	1	3	2	0	1	2	3	3	3	3	3	3	3	3	3	45,38	41,57	33,94
S2	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	43,12	41,83	36,94
S3	3	2	3	3	3	3	3	2	2	2	2	2	2	3	0	3	3	3	1	3	3	51,38	34,42	31,33
S4	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	27,28	34,17	24,29
S5	3	2	2	3	3	3	3	1	3	3	3	1	2	3	3	3	3	3	2	3	2	34,66	48,11	45,35
S6	3	3	3	3	3	3	3	3	3	2	2	1	3	3	3	3	3	3	1	3	1	39,28	42,62	47,72
S7	3	3	3	3	3	3	3	3	3	2	3	3	0	1	3	1	3	3	1	3	1	40,87	33,56	25,95
S8	3	3	3	3	3	3	2	0	1	1	3	0	3	3	3	3	3	3	3	3	1	47,14	37,43	32,72
S9	3	3	3	3	3	2	2	3	1	0	2	2	0	3	0	3	3	3	1	3	2	39,71	33,78	26,63
S10	3	3	3	3	3	3	2	0	3	0	2	0	3	3	1	3	3	3	2	1	1	32,35	32,2	29,31
Ergebnis	30	28	29	30	30	29	25	21	24	16	24	17	22	28	22	28	30	30	20	28	20	40,117	37,969	33,418

Table A.3.: CapTap questionnaire results

Subject	G1	G2	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11
S1	10	5	6	9	9	8	8	1	6	10	9	unsichtbare Sensorik im vertrauten Möbelstück, Gesten über Tisch	Präzision nicht gut genug
S2	10	9	6	8	6	8	7	3	3	7	6	Nice idea of embedding interaction into everyday furniture	area is too large, interaction is exhausting, demonstrator table not suitable
S3	10	4	10	10	5	10	10	1	1	10	7	tactility, intuitive and invisible	some areas hard to reach when sitting in front
S4	10	8	8	9	8	9	2	1	4	10	8		some errors in interaction
S5	10	3	10	8	8	8	2	1	6	9	8	touchpad like control on table	knocking, interaction disturbed by knee
S6	10	8	8	9	7	8	3	3	3	9	8	unobtrusive integration in furniture, intuitive interaction	delay in recognition
S7	10	10	10	10	5	10	3	2	5	10	8	Eingaben mit und ohne Kontakt, Auswahl über Tap intuitiv, Schnelle Eingewöhnung in Layer mit Farben	Beine werden erkannt, große Interaktionsarea, Verzerrungen im Zeichenprogramm
S8	10	3	7	8	7	9	2	2	1	7	4	ease of use, low learning curve, big variety of input modalities	interaction less precise in some areas
S9	10	3	8	9	4	7	7	10	3	9	3	knockin on heaven's door	tap didn't work - required more than one tap to recognize
S10	4	7	8	9	7	9	2	2	7	10	8	different interaction possibilities	detection of knees, difficulty to stay in layer
Ergebnis	9,4	6	8,1	8,9	6,6	8,6	4,6	2,6	3,9	9,1	6,9		

A.4. Active Armrest evaluation results

A.4.1. Questionnaire

- G1 Experience with touch screen systems (none = 1, daily usage = 10)
- G2 Experience with gesture recognition systems (none = 1, daily usage = 10)
- Q1 Do you agree that the required tasks were easy and precise to perform? (Not agree = 1, Strong agree = 10)
- Q2 Was the Interactive Armrest to be intuitive in its usage? (Not agree = 1, Strong agree = 10)
- Q3 Do you prefer multi-touch or basic gestures? (Multi-touch = 1, Basic gesture = 10)
- Q4 Is the Interactive Armrest with Multi-touch easy to use? (Not Agree = 1, Strong Agree = 10)
- Q5 Is the Interactive Armrest with basic gestures easy to use? (Not Agree = 1, Strong Agree = 10)
- Q6 Do you agree that the Interactive Armrest is an interesting form of interaction device? (Not agree = 1, Strong agree = 10)
- Q7 Would you consider using an input device like the Interactive Armrest for a longer period of time? (Not at all = 1, I would like to use it = 10)
- Q8 What did you particularly like about the Interactive Armrest?
- Q9 What did you particularly dislike about the Interactive Armrest?

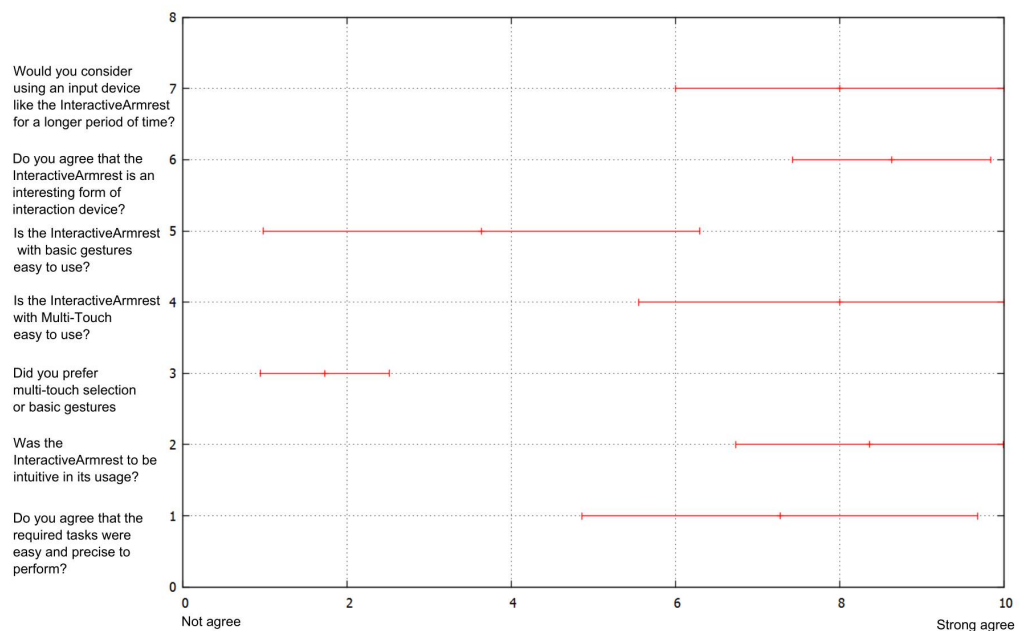


Figure A.5.: Mean values and standard deviations for the Likert items of Active Armrest questionnaire

A.5. CapFloor @ EvAAL 2011

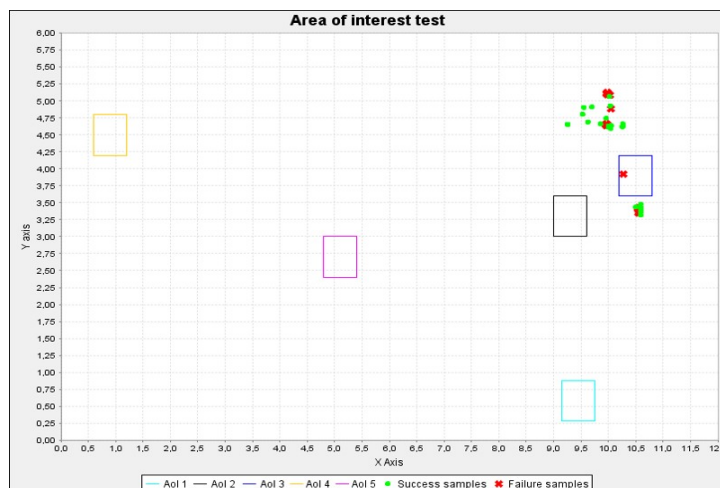


Figure A.6.: Recognition rate of CapFloor on selected areas of interest

Table A.4.: Best case scores of CapFloor @ EvAAL 2011

Availability		
	Run 1	Run 2
Number of wrong samples	210	258
Number of total received samples	328	437
T	0.3597561	0.40961098
Accuracy Score	3.59756098	4.09610984
Accuracy		
Available samples	358	437
Expected samples	452	452
Availability	0.7920354	0.96681416
Availability score	7.92035398	9.66814159

A.6. Smart Bed sleep phase recognition results

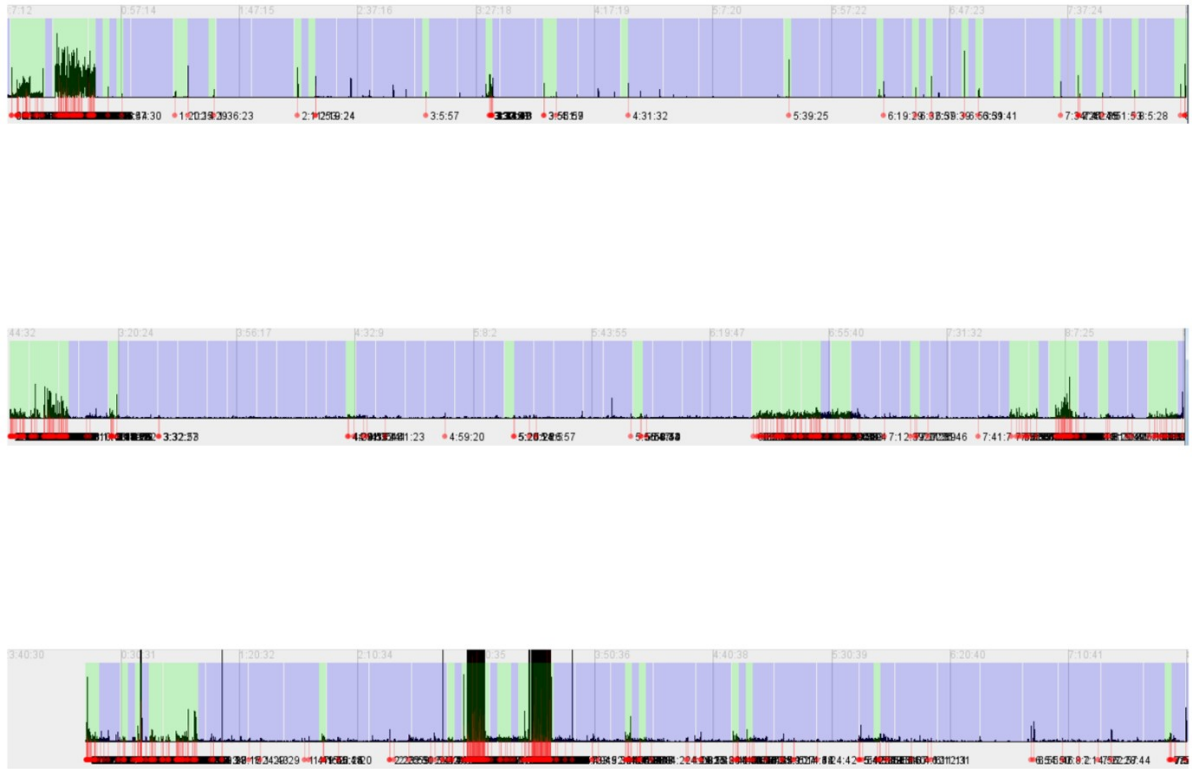


Figure A.7.: Recognition of sleep phases over three nights

B. Publications

The thesis is partially based on the following publications:

Refereed Articles

- 2009 Using the human body field as a medium for natural interaction** (Andreas Braun, Pascal Hamisu), *In Proceedings of the 2nd International Conference on Pervasive Technologies Related to Assistive Environments - PETRA '09*, ACM Press, pp. 1-7, 2009.
- 2010 Analyse des Schlafverhaltens durch kapazitive Sensorarrays zur Ermittlung der Wirbelsäulenbelastung** (Pascal Hamisu, Andreas Braun), *In 3. Deutscher AAL Kongress*, VDE Verlag GmbH, pp. 3-6, 2010.
- 2011 Empowering and integrating senior citizens with virtual coaching** (Andreas Braun, Peter H. M. P. Roelofsma, Dieter Ferring, Milla Immonen), *In Ambient Intelligence (David V. Keyson, Mary Lou Maher, Norbert Streitz, Adrian Cheok, Juan Carlos Augusto, Reiner Wichert, Gwenn Englebienne, Hamid Aghajan, Ben J. A. Kröse, eds.)*, Springer Berlin Heidelberg, volume 7040, pp. 369-370, 2011.
- 2011 Interactive personalization of ambient assisted living environments** (Alexander Marinc, Carsten Stocklöh, Andreas Braun, Carsten Limberger, Cristian Hofmann, Arjan Kuijper), *In Proceeding HI'11 Proceedings of the 2011 international conference on Human interface and the management of information*, volume Part I, pp. 567-576, 2011.
- 2011 Passive identification and control of arbitrary devices in smart environments** (Andreas Braun, Felix Kamieth), *In HCII'11 Proceedings of the 14th international conference on Human-computer interaction: towards mobile and intelligent interaction environments (Julie A. Jacko, ed.)*, Springer-Verlag, pp. 147-154, 2011.
- 2011 Adaptive implicit interaction for healthy nutrition and food intake supervision** (Felix Kamieth, Andreas Braun, Christian Schlehuber), *In HCII'11 Proceedings of the 14th international conference on Human-computer interaction: towards mobile and intelligent interaction environments*, Springer-Verlag, pp. 205-212, 2011.
- 2011 Classification of User Postures with Capacitive Proximity Sensors in AAL-Environments** (Tobias Grosse-Puppenthal, Alexander Marinc, Andreas Braun), *In Ambient Intelligence (David V Keyson, Mary Lou Maher, Norbert Streitz, Adrian Cheok, Juan Carlos Augusto, Reiner Wichert, Gwenn Englebienne, Hamid Aghajan, Ben J A Kröse, eds.)*, Springer, volume 7040, pp. 314-323, 2011.
- 2011 Designing a multi-purpose capacitive proximity sensing input device** (Andreas Braun, Pascal Hamisu), *In Proceedings of the 4th International Conference on Pervasive Technologies Related to Assistive Environments PETRA 11*, ACM Press, 2011.
- 2012 Dynamic User Representation in Video Phone Applications** (Andreas Braun, Reiner Wichert), *In Constructing Ambient Intelligence*, Springer Berlin Heidelberg, pp. 184-188, 2012.

- 2012 Honeyfish - a high resolution gesture recognition system based on capacitive proximity sensing** (Tobias Grosse-Puppenthal, Andreas Braun), *In Embedded World Conference 2012*, Haar: WEKA Fachmedien, 2012 (Design & Elektronik), pp. 10-14, 2012.
- 2012 Context recognition using capacitive sensor arrays in beds** (Andreas Braun, Henning Heggen), *In Technik für ein selbstbestimmtes Leben - 5. Deutscher AAL-Kongress*, VDE VERLAG GmbH, 2012.
- 2012 Visual Support System for Selecting Reactive Elements in Intelligent Environments** (Martin Majewski, Andreas Braun, Alexander Marinc, Arjan Kuijper), *In International Conference on Cyberworlds*, pp. 251-255, 2012.
- 2012 CapFloor – A Flexible Capacitive Indoor Localization System** (Andreas Braun, Henning Heggen, Reiner Wichert), *In Evaluating AAL Systems Through Competitive Benchmarking. Indoor Localization and Tracking* (Stefano Chessa, Stefan Knauth, eds.), Communications in Computer and Information Science, pp. 26-35, 2012.
- 2012 Synergieeffekte aus der Kombination verschiedener AAL Lösungen** (Kerstin Klauß, Stefanie Müller, Andreas Braun, Tim Dutz, Felix Kamieth, Peter Klein), *In Mensch & Computer 2012–Workshopband: interaktiv informiert—allgegenwärtig und allumfassend!?* (H. Reiterer, O. Deussen, eds.), Oldenbourg Verlag, pp. 61-67, 2012.
- 2012 V2me: Evaluating the first steps in mobile friendship coaching** (Salla Muuraiskangas, Anja K Leist, Andreas Braun, Kerstin Klauß, Peter H M P Roelofsma, Reiner Wichert, Peter Klein, Dieter Ferring), *In Journal of Ambient Intelligence and Smart Environments*, IOS Press, volume 4, pp. 517-534, 2012.
- 2013 OpenCapSense: A Rapid Prototyping Toolkit for Pervasive Interaction Using Capacitive Sensing** (Tobias Grosse-Puppenthal, Yannick Berghoefer, Andreas Braun, Raphael Wimmer, Arjan Kuijper), *In IEEE International Conference on Pervasive Computing and Communications (PerCom)*, volume 18, pp. 22, 2013.
- 2013 User requirements for navigation assistance in public transit for elderly people** (Stefanie Müller, Felix Kamieth, Andreas Braun, Tim Dutz, Peter Klein), *In Proceedings of the 6th International Conference on Pervasive Technologies Related to Assistive Environments*, pp. 55, 2013.
- 2013 Swiss-cheese extended: an object recognition method for ubiquitous interfaces based on capacitive proximity sensing** (Tobias Grosse-Puppenthal, Andreas Braun, Felix Kamieth, Arjan Kuijper), *In Proceedings of the 2013 ACM annual conference on Human factors in computing systems*, pp. 1401-1410, 2013.
- 2013 Context-based bounding volume morphing in pointing gesture applications** (Andreas Braun, Arthur Fischer, Alexander Marinc, Carsten Stocklów, Martin Majewski), *In HCII'13 Proceedings of the 15th international conference on Human-computer interaction*, Springer Berlin Heidelberg, pp. 147-156, 2013.
- 2013 Capacitive sensor-based hand gesture recognition in ambient intelligence scenarios** (Andreas Braun, Tim Dutz, Felix Kamieth), *In Proceedings of the 6th International Conference on Pervasive Technologies Related to Assistive Environments*, pp. 5, 2013.
- 2013 Marker-Free Indoor Localization and Tracking of Multiple Users in Smart Environments Using a Camera-Based Approach** (Andreas Braun, Tim Dutz, Michael Alekseew, Philipp Schillinger, Alexander Marinc), *In Distributed, Ambient and Pervasive Interactions*, 2013.
- 2013 Building Up Virtual Environments Using Gestures** (Alexander Marinc, Carsten Stocklów, Andreas Braun), *In Universal Access in Human-Computer Interaction. Applications and Services for Quality of Life*, Springer Berlin Heidelberg, pp. 70-78, 2013.

-
- 2013 Providing Visual Support for Selecting Reactive Elements in Intelligent Environments** (Martin Majewski, Andreas Braun, Alexander Marinc, Arjan Kuijper), *In Transactions on Computational Science XVIII*, Springer, pp. 248-263, 2013.
- 2013 Unobtrusive Recognition of Working Situations** (Tobias Grosse-Puppenthal, Sebastian Benchea, Felix Kamieth, Andreas Braun, Christian Schuster), *In Distributed, Ambient, and Pervasive Interactions*, Springer Berlin Heidelberg, pp. 115-121, 2013.
- 2013 AmbiTrack-Marker-free Indoor Localization and Tracking of Multiple Users in Smart Environments with a Camera-based Approach** (Andreas Braun, Tim Dutz), *In Evaluating AAL Systems Through Competitive Benchmarking*, Springer Berlin Heidelberg, pp. 83-93, 2013.
- 2014 V2me - Virtual Coaching for Seniors** (Andreas Braun, Silvana Cieslik, René Zmugg, Reiner Wichert, Peter Klein, Sven Havemann), *In Wohnen – Pflege – Teilhabe - Besser leben durch Technik. 7. Deutscher AAL-Kongress*, VDE Verlag, 2014.
- 2014 A Gesture-Based Door Control Using Capacitive Sensors** (Steeven Zeiß, Alexander Marinc, Andreas Braun, Tobias Große-Puppenthal, Sebastian Beck), *In Distributed, Ambient, and Pervasive Interactions*, Springer International Publishing, pp. 207-216, 2014.
- 2014 Robot-Supported Pointing Interaction for Intelligent Environments** (Mark Prediger, Andreas Braun, Alexander Marinc, Arjan Kuijper), *In Distributed, Ambient, and Pervasive Interactions*, Springer International Publishing, pp. 172-183, 2014.
- 2014 MoviBed - sleep analysis using capacitive sensors** (Maxim Djakow, Andreas Braun, Alexander Marinc), *In Universal Access in Human-Computer Interaction. Design for All and Accessibility Practice*, Springer International Publishing, pp. 171-181, 2014.

In Press

- 2014 Towards interactive car interiors - the Active Armrest** (Andreas Braun, Stephan Neumann, Sönke Schmidt, Reiner Wichert, Arjan Kuijper), *In Proceedings 8th Nordic conference on human-computer interaction NordiCHI 2014 - Extended Abstracts*, ACM Press, 2014.
- 2014 A benchmarking model for sensors in smart environments** (Andreas Braun, Reiner Wichert, Arjan Kuijper, Dieter W. Fellner), *In Proceedings Aml 14 - European Conference on Ambient Intelligence*, Springer International Publishing, 2014.
- 2015 Capacitive proximity sensing in smart environments** (Andreas Braun, Reiner Wichert, Arjan Kuijper, Dieter W. Fellner), *In Journal of Ambient Intelligence and Smart Environments*, IOS Press, 2015.

Working Papers

- 2014 The Capacitive Chair** (Andreas Braun, Sebastian Frank, Reiner Wichert), 2014.
- 2014 CapTap - Combining Capacitive Gesture Recognition and Knock Detection** (Andreas Braun, Sebastian Zander-Walz, Stefan Krepp, Silvia Rus, Reiner Wichert, Arjan Kuijper), 2014.
- 2014 Low-cost Indoor Localization Using Cameras - Evaluating AmbiTrack and its Applications in Ambient Assisted Living** (Andreas Braun, Tim Dutz), 2014.

C. Supervising Activities

The following list summarizes the student bachelor, diploma and master thesis supervised by the author. The results of these works were partially used as an input into the thesis.

Diploma and Master Thesis

1. Große-Puppendahl, Tobias - Multi-hand Interaction Using Custom Capacitive Proximity Sensors - MSc TU Darmstadt 2012
2. Berghöfer, Yannick - Human-Machine-Interfaces in Automotive Environments using Capacitive Proximity Sensors - MSc TU Darmstadt 2013
3. Krepp, Stefan - Unobtrusive Surface Touch Recognition using Acoustic Tracking - MSc TU Darmstadt 2014
4. Frank, Sebastian - Vehicle-Integrated System for Personal Health Management - MSc Hochschule Rhein-Main 2014

Bachelor Thesis

1. Fischer, Arthur - Unterstützung von zielbasierter Interaktion durch gestenerkennende Zeigegeräte - BSc TU Darmstadt 2012
2. Majewski, Martin - Visual-aided Selection of Reactive Elements in Intelligent Environments - BSc TU Darmstadt 2012
3. Neumann, Stephan - Automotive interfaces using an interactive armrest - BSc TU Darmstadt 2014
4. Zander-Walz, Sebastian - Curved large-area surfaces for gestural interaction - BSc TU Darmstadt 2014

D. Curriculum Vitae

Personal Data

Name	Andreas Braun
Birth date & place	04.10.1982 Aschaffenburg, Germany
Nationality	German

Education

2008 – 2010	Master of Science in Computational Engineering at Technical University of Darmstadt, Germany
2004 – 2008	Bachelor of Science in Computational Engineering at Technical University of Darmstadt, Germany
2002 – 2004	Study of Physics at Julius-Maximilians Universität in Würzburg, Germany

Work Experience

2010 –	Researcher, Competence Center Interactive Multimedia Appliances, Fraunhofer Institute for Computer Graphics Research, Darmstadt, Germany, Focus: HCI applications in smart environments
2008 – 2009	Student Assistant, Competence Center Interactive Multimedia Appliances, Fraunhofer Institute for Computer Graphics Research, Darmstadt, Germany, Focus: Sensor applications and interactive systems

Bibliography

- [AAB98] ARMSTRONG D., ASHER J., BENITEZ S.: Surface acoustic wave touchscreen with housing seal, 1998. 40
- [AGWF11] ANNETT M., GROSSMAN T., WIGDOR D., FITZMAURICE G.: Medusa: a proximity-aware multi-touch tabletop. In *Proceedings UIST* (2011), ACM, pp. 337–346. 110
- [AKKM14] ADIB F., KABELAC Z., KATABI D., MILLER R. C.: 3D Tracking via Body Radio Reflections. *Proceedings of the 11th USENIX Conference on Networked Systems Design and Implementation* (2014), 317–329. xiv, 36, 37
- [AWD*13] ARNDT M., WILLE S., DE SOUZA L., REY V. F., WEHN N., BERNS K.: Performance evaluation of ambient services by combining robotic frameworks and a smart environment platform. *Ubiquitous Robotics* 61, 11 (2013), 1173–1185. 38
- [Bax96] BAXTER L. K.: *Capacitive Sensors*. 1996. 7, 17, 18
- [BB98] BERTOZZI M., BROGGI A.: GOLD: A parallel real-time stereo vision system for generic obstacle and lane detection. *IEEE Transactions on Image Processing* 7, 1 (1998), 62–81. 56
- [BBL93] BAUDEL T., BEAUDOUIN-LAFON M.: Charade: remote control of objects using free-hand gestures. *Communications of the ACM* 36, 7 (July 1993), 28–35. 107, 109
- [BBS06] BALLAGAS R., BORCHERS J., ROHS M., SHERIDAN J. G.: The smart phone: a ubiquitous input device. *IEEE Pervasive Computing* 5, 1 (2006), 70–77. 38, 44
- [BD13] BRAUN A., DUTZ T.: AmbiTrack-Marker-free Indoor Localization and Tracking of Multiple Users in Smart Environments with a Camera-based Approach. In *Proceedings Evaluating AAL Systems Through Competitive Benchmarking*. 2013, pp. 83–93. 38, 39
- [BDA*13] BRAUN A., DUTZ T., ALEKSEEW M., SCHILLINGER P., MARINC A.: Marker-Free Indoor Localization and Tracking of Multiple Users in Smart Environments Using a Camera-Based Approach. In *Proceedings DAPI* (2013), pp. 349–357. xiv, 39
- [BDK13] BRAUN A., DUTZ T., KAMIETH F.: Capacitive sensor-based hand gesture recognition in ambient intelligence scenarios. In *Proceedings PETRA* (2013), ACM, p. Article No. 5. vii, xiv, xv, 67, 68, 104, 105, 106
- [BF14] BRAUN A., FRANK S.: The Capacitive Chair. Working Paper, 2014. vii, 73, 85, 95, 96
- [BGS*06] BROXVALL M., GRITTI M., SAFFIOTTI A., SEO B.-S., CHO Y.-J.: PEIS ecology: Integrating robots into smart environments. In *Proceedings IEEE International Conference on Robotics and Automation* (2006), IEEE, pp. 212–218. xiv, 38, 45, 46
- [BH11] BRAUN A., HAMISU P.: Designing a multi-purpose capacitive proximity sensing input device. In *Proceedings PETRA* (2011), p. Article No. 15. xiv, xv, 67, 104, 105
- [BH12] BRAUN A., HEGGEN H.: Context recognition using capacitive sensor arrays in beds. In *Proceedings AAL-Kongress* (2012). vii, xiv, xv, 71, 72, 73, 92, 93
- [BHK97] BELHUMEUR P. N., HESPANHA J. A. P., KRIEGMAN D. J.: Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Transactions on Pattern Analysis and Machine*

- Intelligence* 19, 7 (1997), 711–720. 17
- [BHW12] BRAUN A., HEGGEN H., WICHERT R.: CapFloor – A Flexible Capacitive Indoor Localization System. In *Proceedings Evaluating AAL Systems Through Competitive Benchmarking. Indoor Localization and Tracking* (2012), pp. 26–35. vii, xv, 68, 89, 91
- [BI04] BAO L., INTILLE S. S.: Activity Recognition from User-Annotated Acceleration Data. In *Proceedings PERVASIVE* (2004), pp. 1–17. 38, 43
- [BJ98] BUCZAK A., JAMALABAD V.: Self-organization of a heterogeneous sensor network by genetic algorithms. *Intelligent Engineering Systems Through Artificial Neural Networks* 8 (1998), 259–264. 76
- [BKKF13] BECK S., KUNERT A., KULIK A., FROEHLICH B.: Immersive Group-to-Group Telepresence. *IEEE Transactions on Visualization and Computer Graphics* 19, 4 (2013), 616–625. 32
- [BNLH11] BRAGDON A., NELSON E., LI Y., HINCKLEY K.: Experimental analysis of touch-screen gesture designs in mobile environments. In *Proceedings CHI* (2011), ACM, pp. 403–412. 102
- [BNS*14] BRAUN A., NEUMANN S., SCHMIDT S., WICHERT R., KUIJPER A.: Towards Interactive Car Interiors - the Active Armrest. In *Proceedings NordiCHI Extended Abstracts* (2014), ACM. In press. vii, xv, 77, 101, 102
- [BO10a] BARRETT G., OMOTE R.: Projected capacitive touch screens. *Information Display* (2010). xiii, 12, 68
- [BO10b] BARRETT G., OMOTE R.: Projected-capacitive touch technology. *Information Display* (2010). 9, 11
- [BWGT11] BULLING A., WARD J. A., GELLERSEN H., TROSTER G.: Eye movement analysis for activity recognition using electrooculography. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33, 4 (2011), 741–753. xiv, 38, 42, 43
- [BWKf14] BRAUN A., WICHERT R., KUIJPER A., FELLNER D. W.: A benchmarking model for sensors in smart environments. In *Proceedings Aml International* (2014), Springer International Publishing. In press. 47
- [BWKf15] BRAUN A., WICHERT R., KUIJPER A., FELLNER D. W.: Capacitive proximity sensing in smart environments. *Journal of Ambient Intelligence and Smart Environments* (2015). In Press. 7, 125
- [BZWK*14] BRAUN A., ZANDER-WALZ S., KREPP S., RUS S., WICHERT R., KUIJPER A.: CapTap - Combining Capacitive Gesture Recognition and Knock Detection. Working Paper, 2014. viii, 80, 81, 107, 109
- [CAL10] CHENG J., AMFT O., LUKOWICZ P.: Active capacitive sensing: Exploring a new wearable sensing modality for activity recognition. In *Pervasive Computing*. Springer, 2010, pp. 319–336. 61, 65, 84, 120, 121
- [CC89] CAMP R. C., CAMP ROBERT C.: Benchmarking: the search for industry best practices that lead to superior performance. 47
- [CC13] CYPRESS SEMICONDUCTOR CORPORATION, CORPORATION C. S.: CY3271 PSoC® FirstTouch™ Starter Kit with CYFi™ Low-Power RF. <http://www.cypress.com/?rID=17672>, 2013. Accessed: 2014-07-22. 105
- [CD07] COOK D. J., DAS S. K.: How smart are our environments? An updated look at the state of the art. *Pervasive and mobile computing* 3, 2 (2007), 53–73. xvii, 3, 27, 28, 39

-
- [CDCT*01] COWIE R., DOUGLAS-COWIE E., TSAPATSOULIS N., VOTSIS G., KOLLIAS S., FELLEZ W., TAYLOR J. G.: Emotion recognition in human-computer interaction. *IEEE Signal Processing Magazine* 18, 1 (2001), 32–80. 38, 41
 - [CHE00] CRAWFORD L. E., HUTTENLOCHER J., ENGBRETSON P. H.: Category effects on estimates of stimuli: Perception or reconstruction? *Psychological Science* 11, 4 (2000), 280–284. 56
 - [CK12] CHessa S., KNAUTH S.: Evaluating AAL Systems Through Competitive Benchmarking. In *Indoor Localization and Tracking*. In *Proceedings Evaluating AAL Systems Through Competitive Benchmarking. Indoor Localization and Tracking* (2012), pp. 1–13. 39, 48, 90
 - [Cla62] CLARKE A. C.: Hazards of prophecy: The failure of imagination. *Profiles of the Future, Gollancz, London* (1962). 2
 - [CLC*06] CHANG K.-H., LIU S.-Y., CHU H.-H., HSU J. Y.-J., CHEN C., LIN T.-Y., CHEN C.-Y., HUANG P.: The diet-aware dining table: Observing dietary behaviors over a tabletop surface. In *Pervasive Computing*. Springer, 2006, pp. 366–382. 80
 - [CPP10] CHINTALAPUDI K., PADMANABHA IYER A., PADMANABHAN V. N.: Indoor localization without the pain. In *Proceedings MobiCom* (2010), ACM, pp. 173–184. 38, 39
 - [Cro09] CROLOTTE A.: Issues in benchmark metric selection. In *Performance Evaluation and Benchmarking*. Springer, 2009, pp. 146–152. 48
 - [CRV08] CORBISHLEY P., RODRÍGUEZ-VILLEGAS E.: Breathing detection: towards a miniaturized, wearable, battery-operated monitoring system. *IEEE Transactions on Biomedical Engineering* 55, 1 (2008), 196–204. 34
 - [CS06] CORADESCHI S., SAFFIOTTI A.: Symbiotic robotic systems: Humans, robots, and smart environments. *IEEE Intelligent Systems* 21, 3 (2006), 82–84. 38, 45
 - [Cyp12] CYPRESS: Cypress TrueTouch® Touchscreen Solution Drives "Floating Touch" Navigation Feature in New Xperia™ sola Smartphone from Sony Mobile Communications. Tech. rep., 2012. 9
 - [DBM14] DJAKOW M., BRAUN A., MARINC A.: MoviBed - sleep analysis using capacitive sensors. In *Proceedings UAHCI* (2014), pp. 171–181. vii, xv, 86, 92, 93, 94
 - [DGT*13] DEMENTYEV A., GUMMESON J., THRASHER D., PARKS A., GANESAN D., SMITH J. R., SAMPLE A. P.: Wirelessly powered bistable display tags. In *Proceedings UbiComp* (2013), ACM, pp. 383–386. 38, 44
 - [DL01] DIETZ P., LEIGH D.: DiamondTouch. In *Proceedings UIST* (Nov. 2001), pp. 219–226. 12
 - [DLP03] DONGARRA J. J., LUSZCZEK P., PETITET A.: The LINPACK benchmark: past, present and future. *Concurrency and Computation: practice and experience* 15, 9 (2003), 803–820. 47
 - [DT09] DEARMAN D., TRUONG K. N.: BlueTone: a framework for interacting with public displays using dual-tone multi-frequency through bluetooth. In *Proceedings UbiComp* (2009), ACM, pp. 97–100. 38, 44
 - [Fra13] FRANK R.: *Understanding smart sensors*. Artech House, 2013. 27
 - [GBK99] GELLERSEN H.-W., BEIGL M., KRULL H.: The MediaCup: Awareness technology embedded in an everyday object. In *Proceedings Handheld and Ubiquitous Computing* (1999), Springer, pp. 308–310. xiv, 38, 43, 44
 - [Get11] GETREUER P.: Linear Methods for Image Interpolation. *IPOL Journal - Image Processing On Line* 1 (2011). 83
-

- [GKIH09] GLAS D. F., KANDA T., ISHIGURO H., HAGITA N.: Simultaneous people tracking and localization for social robots using external laser range finders. In *Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems* (2009), IEEE, pp. 846–853. xiv, 46
- [Gli00] GLINSKY A.: *Theremin: Ether music and espionage*. University of Illinois Press, 2000. v, xiii, 19
- [GPB12] GROSSE-PUPPENDAHL T., BRAUN A.: Honeyfish - a high resolution gesture recognition system based on capacitive proximity sensing. In *Proceedings Embedded World Conference* (2012), pp. 10–14. 117, 129
- [GPBB12] GROSSE-PUPPENDAHL T., BERLIN E., BORAZIO M.: Enhancing accelerometer-based activity recognition with capacitive proximity sensing. In *Proceedings Aml International*. 2012, pp. 17–32. 61
- [GPBB*13] GROSSE-PUPPENDAHL T., BERGHOFER Y., BRAUN A., WIMMER R., KUIJPER A.: Open-CapSense: A Rapid Prototyping Toolkit for Pervasive Interaction Using Capacitive Sensing. *Proceedings PerCom* (2013), 152 – 159. xiii, 11, 59, 108, 111
- [GPBKK13] GROSSE-PUPPENDAHL T., BRAUN A., KAMIETH F., KUIJPER A.: Swiss-cheese extended: an object recognition method for ubiquitous interfaces based on capacitive proximity sensing. In *Proceedings CHI* (2013), ACM, pp. 1401–1410. 17, 59, 117, 137
- [GPMB11] GROSSE-PUPPENDAHL T., MARINC A., BRAUN A.: Classification of User Postures with Capacitive Proximity Sensors in AAL-Environments. In *Proceedings Aml International* (2011), vol. 7040, pp. 314–323. xiii, 12, 13, 116
- [GZBB09] GEORGE B., ZANGL H., BRETTERKLIEBER T., BRASSEUR G.: Seat occupancy detection based on capacitive sensing. *IEEE Transactions on Instrumentation and Measurement* 58, 5 (2009), 1487–1494. 18
- [Han05] HAN J. Y.: Low-cost multi-touch sensing through frustrated total internal reflection. In *Proceedings UIST* (Oct. 2005), pp. 115–118. xiv, 40
- [HB10] HAMISU P., BRAUN A.: Analyse des Schlafverhaltens durch kapazitive Sensorarrays zur Ermittlung der Wirbelsäulenbelastung. In *Proceedings AAL-Kongress* (2010), pp. 3–6. 71, 92
- [HBvdH*11] HUIJNEN C., BADII A., VAN DEN HEUVEL H., CALEB-SOLLY P., THIEMERT D.: “Maybe it becomes a buddy, but do not call it a robot”—seamless cooperation between companion robotics and smart homes. In *Proceedings Aml International*. 2011, pp. 324–329. 38, 46
- [HCM*13] HOQUE M. E., COURGEON M., MARTIN J.-C., MUTLU B., PICARD R. W.: Mach: My automated conversation coach. In *Proceedings UbiComp* (2013), ACM, pp. 697–706. xiv, 38, 41, 42
- [HDO*98] HEARST M. A., DUMAIS S. T., OSMAN E., PLATT J., SCHOLKOPF B.: Support vector machines. *IEEE Intelligent Systems and their Applications* 13, 4 (1998), 18–28. 75, 81
- [Hen00] HENNING J. L.: SPEC CPU2000: Measuring CPU performance in the new millennium. *Computer* 33, 7 (2000), 28–35. 48
- [HH08] HARRISON C., HUDSON S. E.: Scratch input: creating large, inexpensive, unpowered and mobile finger input surfaces. In *Proceedings UIST* (2008), ACM, pp. 205–208. 80
- [HPGK94] HINCKLEY K., PAUSCH R., GOBLE J. C., KASSELL N. F.: A survey of design issues in spatial input. In *Proceedings UIST* (1994), ACM, pp. 213–222. 77
- [HPR00] HAN J., PRATT M., REGLI W. C.: Manufacturing feature recognition from solid models: a status report. *IEEE Transactions on Robotics and Automation* 16, 6 (2000), 782–796. 17

-
- [HSH11] HARRISON C., SCHWARZ J., HUDSON S. E.: TapSense: enhancing finger interaction on touch surfaces. In *Proceedings UIST* (2011), ACM, pp. 627–636. xiv, 34, 35, 80, 127
 - [HSM00] HARADA T., SATO T., MORI T.: Human motion tracking system based on skeleton and surface integration model using pressure sensors distribution bed. In *Proceedings Workshop on Human Motion* (2000), IEEE, pp. 99–106. 70, 73
 - [HSP12] HARRISON C., SATO M., POUPYREV I.: Capacitive fingerprinting: exploring user differentiation by sensing electrical properties of the human body. In *Proceedings UIST* (2012), ACM, pp. 537–544. xiii, 25
 - [Hu62] HU M.-K.: Visual pattern recognition by moment invariants. *IRE Transactions on Information Theory* 8, 2 (1962), 179–187. 83
 - [iee07] IEEE Standard for a Smart Transducer Interface for Sensors and Actuators Wireless Communication Protocols and Transducer Electronic Data Sheet (TEDS) Formats, 2007. 27
 - [IKH11] IZADI S., KIM D., HILLIGES O.: KinectFusion: real-time 3D reconstruction and interaction using a moving depth camera. In *Proceedings UIST* (2011), pp. 559 – 568. xiv, 31
 - [Jai91] JAIN R.: *The art of computer systems performance analysis*, vol. 182. John Wiley & Sons Chichester, 1991. 48
 - [Jon70] JONES B.: *The Life and Letters of Faraday*, vol. 1. Longmans, Green and Company, 1870. 7
 - [KHC*13] KAWAHARA Y., HODGES S., COOK B. S., ZHANG C., ABOWD G. D.: Instant inkjet circuits: lab-based inkjet printing to support rapid prototyping of UbiComp devices. In *Proceedings UbiComp* (2013), ACM, pp. 363–372. 136
 - [KJJ11] KREJCAR O., JIRKA J., JANCKULIK D.: Use of mobile phones as intelligent sensors for sound input analysis and sleep state detection. *Sensors* 11, 6 (2011), 6037–6055. 92
 - [KMS*12] KOELSTRA S., MUHL C., SOLEYMANI M., LEE J.-S., YAZDANI A., EBRAHIMI T., PUN T., NIJHOLT A., PATRAS I.: Deap: A database for emotion analysis; using physiological signals. *IEEE Transactions on Affective Computing* 3, 1 (2012), 18–31. xiv, 38, 41, 42
 - [KOR*02] KRUM D. M., OMOTESO O., RIBARSKY W., STARNER T., HODGES L. F.: Speech and gesture multimodal control of a whole Earth 3D visualization environment. In *Proceedings VISSYM* (May 2002), pp. 195–200. 107
 - [KQWA10] KHOSROWABADI R., QUEK H. C., WAHAB A., ANG K. K.: EEG-based emotion recognition using self-organizing map for boundary detection. In *Proceedings ICPR* (2010), IEEE, pp. 4242–4245. 38, 41
 - [LBT02] LENMAN S., BRETZNER L., THURESSON B.: Using marking menus to develop command sets for computer vision based hand gesture interfaces. In *Proceedings NordiCHI* (2002), pp. 239–242. 107, 109, 114
 - [LC85] LEWIS B. C., CREWS A. E.: The evolution of benchmarking as a computer performance evaluation technique. *MIS Quarterly* (1985), 7–16. 47
 - [Lew04] LEWIS F. L.: Wireless sensor networks. *Smart environments: technologies, protocols, and applications* (2004), 11–46. 27
 - [LS09] LAUTERBACH C., STEINHAGE A.: SensFloor® - A Large-area Sensor System Based on Printed Textiles Printed Electronics. In *Proceedings AAL-Kongress* (2009). 60, 64, 68, 120, 121
 - [LSKB13] LASECKI W. S., SONG Y. C., KAUTZ H., BIGHAM J. P.: Real-time crowd labeling for deployable activity recognition. In *Proceedings CSCW* (2013), ACM, pp. 1203–1212. 38, 43
-

- [LTI*14] LE GOC M., TAYLOR S., IZADI S., KESKIN C., OTHERS: A Low-cost Transparent Electric Field Sensor for 3D Interaction on Mobile Devices. In *Proceedings CHI* (2014), pp. 3167–3170. v, xiii, 17, 27, 120, 122
- [LWG05] LANDSIEDEL O., WEHRLE K., GÖTZ S.: Accurate prediction of power consumption in sensor networks. In *Proceedings Workshop on Embedded Networked Sensors* (2005), Citeseer. 49
- [LZWV09] LIU J., ZHONG L., WICKRAMASURIYA J., VASUDEVAN V.: uWave: Accelerometer-based personalized gesture recognition and its applications. *Pervasive and Mobile Computing* 5, 6 (2009), 657–675. 38, 41
- [MA07] MITRA S., ACHARYA T.: Gesture recognition: A survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews* 37, 3 (2007), 311–324. 40
- [Mac04] MACLACHLAN R.: Spread Spectrum Capacitive Proximity Sensor. <http://humancond.org/wiki/user/ram/electro/capsense/0main>, 2004. Accessed: 2014-07-22. xvi, 59, 65, 85, 120, 122, 132
- [MBMK13] MAJEWSKI M., BRAUN A., MARINC A., KUIJPER A.: Providing Visual Support for Selecting Reactive Elements in Intelligent Environments. In *Transactions on Computational Science XVIII*, vol. 7848. Springer, 2013, pp. 248–263. 38, 41
- [Mic13] MICROCHIP TECHNOLOGY INC.: *GestIC® Design Guide : Electrodes and System Design MGC3130*. Tech. rep., 2013. xiii, 27
- [Mit07] MITCHELL H. B.: *Multi-sensor data fusion*. Springer, 2007. 15
- [Moo14] MOOG MUSIC: Theremini. <http://www.moogmusic.com/node/92916>, 2014. Accessed: 2014-07-22. xiii, 19
- [MPLK05] MOON J., PARK J., LEE T., KIM Y.: Transparent conductive film based on carbon nanotubes and PEDOT composites. *Diamond and related materials* 14, 11 (2005), 1882–1887. 11
- [MWBS09] MULLONI A., WAGNER D., BARAKONYI I., SCHMALSTIEG D.: Indoor positioning and navigation with camera phones. *IEEE Pervasive Computing* 8, 2 (2009), 22–31. xiv, 38, 39, 40
- [Nok12] NOKIA: *Puremotion Technology White Paper*. Tech. rep., 2012. 9
- [Nov14] NOVINT TECHNOLOGIES INC: Novint Falcon. <http://www.novint.com/index.php/novintfalcon>, 2014. Accessed: 2014-07-22. xiii, 20, 21
- [OHR14] OCHIAI Y., HOSHI T., REKIMOTO J.: Three-dimensional mid-air acoustic manipulation by ultrasonic phased arrays. *PloS one* 9, 5 (2014), e97590. xiv, 32, 33
- [OKLVO12] OLSSON T., KÄRKKÄINEN T., LAGERSTAM E., VENTÄ-OLKKONEN L.: User evaluation of mobile augmented reality scenarios. *Journal of Ambient Intelligence and Smart Environments* 4, 1 (2012), 29–47. 38, 45
- [OL13] OREIFEJ O., LIU Z.: Hon4d: Histogram of oriented 4d normals for activity recognition from depth sequences. In *Proceedings CVPR* (2013), IEEE, pp. 716–723. xiv, 38, 42, 43
- [Pan12] PANGER G.: Kinect in the kitchen: testing depth camera interactions in practical home environments. In *Proceedings CHI Extended Abstracts* (2012), ACM, pp. 1985–1990. vi, xiv, 31, 32
- [PC00] PENTLAND A., CHOUDHURY T.: Face recognition for smart environments. *Computer* 33, 2 (2000), 50–55. xiv, 29
- [PCB00] PRIYANTHA N. B., CHAKRABORTY A., BALAKRISHNAN H.: The cricket location-support system. In *Proceedings MobiCom* (2000), ACM, pp. 32–43. vi, xiv, 32, 33

-
- [PGGP13] PU Q., GUPTA S., GOLLAKOTA S., PATEL S.: Whole-home gesture recognition using wireless signals. In *Proceedings MobiCom* (2013), ACM, pp. 27–38. vi, xiv, 36, 37, 38, 41, 127
- [PJS*13] PHAM C., JACKSON D., SCHÖNING J., BARTINDALE T., PLÖTZ T., OLIVIER P.: FoodBoard: surface contact imaging for food recognition. In *Proceedings UbiComp* (2013), ACM, pp. 749–752. xiv, 30
- [PL13] PIRKL G., LUKOWICZ P.: Resonant magnetic coupling indoor localization system. In *Proceedings Evaluating AAL Systems Through Competitive Benchmarking* (2013), ACM, pp. 59–62. xiv, 38, 39, 40
- [Pla99] PLATT J. C.: Fast training of support vector machines using sequential minimal optimization. *Advances in kernel methods* (1999), 185–208. 75, 79
- [Ple12] PLESSEY SEMICONDUCTORS: *Application Note # 291566 Non-contact ECG measurement using EPIC*. Tech. rep., 2012. 84
- [Pos11] POSLAD S.: *Ubiquitous computing: smart devices, environments and interactions*. Wiley, 2011. 2, 3, 39
- [PSF01] PRASSLER E., SCHOLZ J., FIORINI P.: A robotics wheelchair for crowded public environment. *IEEE Robotics & Automation Magazine* 8, 1 (2001), 38–45. xiv, 32, 33
- [PSLS12] POUPYREV I., SCHOESSLER P., LOH J., SATO M.: Botanicus Interacticus: interactive plants technology. In *Proceedings SIGGRAPH Emerging Technologies* (Aug. 2012), ACM, p. 4. xiii, 25, 120, 121, 129
- [PWLK03] PARK S. H., WON S. H., LEE J. B., KIM S. W.: Smart home—digitally engineered domestic life. *Personal and Ubiquitous Computing* 7, 3-4 (2003), 189–196. 43
- [RAD*09] ROBERTSON M., AMICK III B. C., DERANGO K., ROONEY T., BAZZANI L., HARRIST R., MOORE A.: The effects of an office ergonomics training and chair intervention on worker knowledge, behavior and musculoskeletal risk. *Applied Ergonomics* 40, 1 (2009), 124–135. 75
- [RAMB*05] RANGANATHAN A., AL-MUHTADI J., BIEHL J., ZIEBART B., CAMPBELL R. H., BAILEY B.: Towards a pervasive computing benchmark. In *Proceedings PerCom* (2005), IEEE, pp. 194–198. vi, 48
- [RB13] RÜMELIN S., BUTZ A.: How to make large touch screens usable while driving. In *Proceedings AutomotiveUI* (2013), ACM, pp. 48–55. 101
- [Rek02] REKIMOTO J.: SmartSkin: an infrastructure for freehand manipulation on interactive surfaces. In *Proceedings CHI* (2002), ACM, pp. 113–120. v, xiii, 4, 24, 80, 129, 130, 134
- [SAL06] SUBRAMANIAN S., ALIAKSEYEU D., LUCERO A.: Multi-layer interaction for digital tables. In *Proceedings UIST* (2006), ACM, pp. 269–272. 109
- [SAW94] SCHILIT B., ADAMS N., WANT R.: Context-aware computing applications. In *Proceedings Workshop on Mobile Computing Systems and Applications* (1994), IEEE, pp. 85–90. 2
- [Sch08] SCHULZ H.: Rethinking sleep analysis. *Journal of clinical sleep medicine* 4, 2 (Apr. 2008), 99–103. 86
- [Sea97] SEAYER A. E.: Human Body capacitance: A review. In *Proceedings ESA Annual Meeting* (1997), pp. 16–28. 70
- [SFC*13] SHOTTON J., FITZGIBBON A., COOK M., SHARP T., FINOCCHIO M., MOORE R., KIPMAN A., BLAKE A.: Real-time human pose recognition in parts from single depth images. *Communications of the ACM* 56, 1 (June 2013), 116–124. 31, 73
-

- [SKOM06] SUGANO M., KAWAZOE T., OHTA Y., MURATA M.: Indoor localization system using RSSI measurement of wireless sensor network based on ZigBee standard. *Target* 538 (2006), 50. xiv, 35, 36
- [SL86] SALMI T., LEINONEN L.: Automatic analysis of sleep records with static charge sensitive bed. *Electroencephalography and Clinical Neurophysiology* 64, 1 (1986), 84–87. 86, 87, 88, 93
- [SM95] SCHMANDT C., MULLINS A.: AudioStreamer: exploiting simultaneity for listening. In *Conference companion on Human factors in computing systems* (1995), ACM, pp. 218–219. 21
- [Smi88] SMITH J. E.: Characterizing computer performance with a single number. *Communications of the ACM* 31, 10 (1988), 1202–1206. 48
- [Smi96] SMITH J. R.: Field mice: Extracting hand geometry from electric field measurements. *IBM Systems Journal* 35, 3.4 (1996), 587–608. xiii, 4, 10, 65, 117
- [Smi99] SMITH J. R.: *Electric field imaging*. PhD thesis, Massachusetts Institute of Technology, 1999. v, xiii, 10, 17, 19, 20, 21, 61, 77, 81, 120, 122
- [SN04] SHIREHJINI A. N., NAZARI A.: A novel interaction metaphor for personal environment control: direct manipulation of physical environment based on 3D visualization. *Computers & Graphics* 28, 5 (Oct. 2004), 667–675. xiv, 38, 44, 45
- [SPH12] SATO M., POUPYREV I., HARRISON C.: Touché: enhancing touch interaction on humans, screens, liquids, and everyday objects. In *Proceedings CHI* (May 2012), pp. 483–492. v, xiii, 4, 25, 120, 121, 129, 137
- [SPSS11] SUNG J., PONCE C., SELMAN B., SAXENA A.: Human Activity Detection from RGBD Images. In *Proceedings Plan, Activity, and Intent Recognition* (2011). 31
- [Ste13] STEELCASE INC.: Global Posture Study. <http://www.steelcase.com/en/products/category/seating/task/gesture/pages/global-posture-study.aspx>, 2013. Accessed: 2014-07-22. xv, 76, 77, 95
- [SV01] SCHMIDT A., VAN LAERHOVEN K.: How to build smart appliances? *IEEE Personal Communications* 8, 4 (2001), 66–71. 43, 64
- [SWD*98] SMITH J., WHITE T., DODGE C., PARADISO J., GERSHENFELD N., ALLPORT D.: Electric field sensing for graphical interfaces. *IEEE Computer Graphics and Applications* 18, 3 (1998), 54–60. xiii, 21, 65, 70, 81, 134
- [TCY*11] TSAI P.-H., CHEN T.-Y., YU C.-R., SHIH C.-S., LIU J. W. S.: Smart medication dispenser: design, architecture and implementation. *IEEE Systems Journal* 5, 1 (2011), 99–110. 38, 44
- [TKA06] TABAR A. M., KESHAVARZ A., AGHAJAN H.: Smart home care network using sensor fusion and distributed vision-based reasoning. In *Proceedings of the 4th international workshop on Video surveillance and sensor networks* (2006), ACM, pp. 145–154. xiv, 28, 29
- [TSPM01] TAN H. Z., SLIVOVSKY L. A., PENTLAND A., MEMBER S.: A sensing chair using pressure distribution sensors. *IEEE/ASME Transactions on Mechatronics* 6, 3 (2001), 261–268. 95
- [Val08] VALLI A.: The design of natural interaction. *Multimedia Tools and Applications* 38, 3 (2008), 295–305. 65
- [VCBE07] VAJK T., COULTON P., BAMFORD W., EDWARDS R.: Using a mobile phone as a “wii-like” controller for playing games on a large public display. *International Journal of Computer Games Technology* 2008 (2007). xiv, 38, 44, 45

-
- [VKMV11] VALTONEN M., KAILA L., MÄENTAUSTA J., VANHALA J.: Unobtrusive human height and posture recognition with a capacitive sensor. *Journal of Ambient Intelligence and Smart Environments* 3, 4 (2011), 305–332. xiii, 26
 - [VMV09] VALTONEN M., MAENTAUSTA J., VANHALA J.: TileTrack: Capacitive human tracking using floor tiles. In *Proceedings PerCom* (Mar. 2009), pp. 1–10. v, 13, 26, 60, 64, 68, 120, 121
 - [VRV10] VALTONEN M., RAULA H., VANHALA J.: Human body tracking with electric field ranging. In *Proceedings MindTrek* (2010), ACM, pp. 183–186. xiii, 26, 129
 - [VV12] VALTONEN M., VUORELA T.: Capacitive User Tracking Methods for Smart Environments. *InTech, New Approach in Indoor and Outdoor Localization Systems* (2012), 113–138. xiii, 26
 - [Wad54] WADE O. L.: Movements of the thoracic cage and diaphragm in respiration. *The Journal of physiology* 124, 2 (1954), 193–212. 85
 - [WB95] WELCH G., BISHOP G.: An introduction to the Kalman filter, 1995. 15
 - [WB03] WU M., BALAKRISHNAN R.: Multi-finger and whole hand gestural interaction techniques for multi-user tabletop displays. In *Proceedings UIST* (Nov. 2003), pp. 193–202. 107
 - [WB09] WIMMER R., BORING S.: HandSense: discriminating different ways of grasping and holding a tangible user interface. In *Proceedings TEI* (2009), ACM, pp. 359–362. xiii, 23, 61, 120
 - [WB11] WIMMER R., BAUDISCH P.: Modular and deformable touch-sensitive surfaces based on time domain reflectometry. In *Proceedings UIST* (2011), ACM, pp. 517–526. xiii, 23
 - [Wei84] WEICKER R. P.: Dhrystone: a synthetic systems programming benchmark. *Communications of the ACM* 27, 10 (1984), 1013–1030. 47
 - [Wei91] WEISER M.: The Computer for the 21st Century. *Scientific American* 265, 3 (1991), 94–104. xiv, 2, 30, 31, 44
 - [WF07] WILSON T. V., FENLON W.: How the iPhone Works - Multi-touch systems. <http://electronics.howstuffworks.com/iphone2.htm>, 2007. Accessed: 2014-07-22. 17
 - [WFS83] WILDE-FRENN J., SCHULZ H.: Rate and distribution of body movements during sleep in humans. *Perceptual and motor skills* 56, 1 (1983), 275–283. 85
 - [Whi06] WHITE N.: Windows Experience Index: An In-Depth Look. <http://blogs.windows.com/windows/archive/b/windowsvista/archive/2006/09/22/windows-experience-index-an-in-depth-look.aspx>, 2006. Accessed: 2014-07-22. 48
 - [WHKS06] WIMMER R., HOLLEIS P., KRANZ M., SCHMIDT A.: Thracker - Using Capacitive Sensing for Gesture Recognition. In *Proceedings ICDCSW* (2006), pp. 64–64. xiii, 4, 18, 22, 65, 120
 - [Wik06] WIKIMEDIA COMMONS: A bayer pattern on a sensor in isometric perspective/projection. http://commons.wikimedia.org/wiki/File:Bayer_pattern_on_sensor.svg, 2006. Accessed: 2014-07-22. xiii, 29
 - [Wil04] WILSON J. S.: *Sensor technology handbook*. Elsevier, 2004. 48
 - [WKBS07] WIMMER R., KRANZ M., BORING S., SCHMIDT A.: A Capacitive Sensing Toolkit for Pervasive Activity Detection and Recognition. In *Proceedings PerCom* (2007), pp. 171–180. v, xiii, 22, 67, 111, 122
 - [WKSE11] WACHS J. P., KÖLSCH M., STERN H., EDAN Y.: Vision-based hand-gesture applications. *Communications of the ACM* 54, 2 (2011), 60–71. 102
-

- [WME*10] WÖLLMER M., METALLINO A., EYBEN F., SCHULLER B., NARAYANAN S. S.: Context-sensitive multimodal emotion recognition from speech and facial expression using bidirectional LSTM modeling. In *Proceedings INTERSPEECH* (2010), pp. 2362–2365. xiv, 38, 42
- [WP03] WILSON A., PHAM H.: Pointing in intelligent environments with the worldcursor. In *Proceedings International Conference on Human Computer Interaction* (2003). 41
- [WP10] WILSON J., PATWARI N.: Radio tomographic imaging with wireless networks. *IEEE Mobile Computing* 9, 5 (2010), 621–632. xiv, 35, 36
- [WRS*12] WU H.-Y., RUBINSTEIN M., SHIH E., GUTTAG J., DURAND F., FREEMAN W.: Eulerian video magnification for revealing subtle changes in the world. *ACM Transactions on Graphics* 31, 4 (July 2012), 1–8. vi, xiv, 29, 30
- [WS03] WILSON A., SHAFER S.: XWand: UI for intelligent spaces. In *Proceedings CHI* (2003), pp. 545–552. xiv, 38, 40, 41
- [WTT13] WATANABE H., TERADA T., TSUKAMOTO M.: Ultrasound-based movement sensing, gesture-, and context-recognition. In *Proceedings ISWC* (2013), ACM, pp. 57–64. xiv, 32, 33
- [WYG*09] WRIGHT J., YANG A. Y., GANESH A., SASTRY S. S., MA Y.: Robust face recognition via sparse representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 31, 2 (2009), 210–227. 29
- [XLL*13] XU C., LI S., LIU G., ZHANG Y., MILUZZO E., CHEN Y.-F., LI J., FIRNER B.: Crowd++: unsupervised speaker count with smartphones. In *Proceedings UbiComp* (2013), ACM, pp. 43–52. xiv, 34, 35
- [YDM12] YE J., DOBSON S., MCKEEVER S.: Situation identification techniques in pervasive computing: A review. *Pervasive and Mobile Computing* 8, 1 (2012), 36–66. 42
- [YJS06] YILMAZ A., JAVED O., SHAH M.: Object tracking: A survey. *ACM Computing Surveys* 38, 4 (2006), Article No. 13. 16
- [YNR13] YEO K. P., NANAYAKKARA S., RANSIRI S.: StickEar: making everyday objects respond to sound. In *Proceedings UIST* (2013), ACM, pp. 221–226. xiv, 38, 43, 44
- [ZFBZ08] ZHANG C., FLORÊNCIO D., BA D. E., ZHANG Z.: Maximum likelihood sound source localization and beamforming for directional microphone arrays in distributed meetings. *IEEE Multimedia* 10, 3 (2008), 538–548. vi, xiv, 34
- [Zha12] ZHANG Z.: Microsoft kinect sensor and its effect. *IEEE Multimedia* 19, 2 (2012), 4–10. xiv, 31
- [ZLB*86] ZIMMERMAN T. G., LANIER J., BLANCHARD C., BRYSON S., HARVILL Y.: A hand gesture interface device. In *ACM SIGCHI Bulletin* (May 1986), vol. 17, ACM, pp. 189–192. xiv, 38, 40, 107
- [ZSP*95] ZIMMERMAN T. G., SMITH J. R., PARADISO J. A., ALLPORT D., GERSHENFELD N.: Applying electric field sensing to human-computer interfaces. In *Proceedings CHI* (1995), pp. 280–287. 4, 21